

**Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Electrónica
Maestría en Ciencias de la Electrónica,
Opción en Automatización**



**DISEÑO DE OSCILADORES CAÓTICOS DE ORDEN
ENTERO Y FRACCIONAL USANDO HARDWARE
EMBEBIDO**

Tesis presentada para obtener el título de
Maestro en Ciencias de la Electrónica, Opción en Automatización

Presenta: **ING. DANIEL CLEMENTE LÓPEZ***

Directores de Tesis: **DR. JESÚS MANUEL MUÑOZ PACHECO (FCE-BUAP)**
DRA. OLGA GUADALUPE FÉLIX BELTRÁN (FCE-BUAP)
DR. ESTEBAN TLELO CUAUTLE (INAOE)

*Becario CONACyT

Inicio: agosto 2017
Término: septiembre 2019.

Agradecimientos

En primera instancia, agradezco a la Benemérita Universidad Autónoma de Puebla y a la Facultad de Ciencias de la Electrónica por haberme brindado todo el apoyo durante mi estancia en la Maestría de Ciencias de la Electrónica, Opción en Automatización.

Agradezco a mis asesores de tesis, al Dr. Jesús Manuel Muñoz Pacheco, a la Dra. Olga Félix Beltrán y al Dr. Esteban Tlelo Cuautle que, por sus conocimientos, orientaciones y responsabilidad me ayudaron a culminar este trabajo.

Extiendo mi agradecimiento a los jurados de tesis, la Dra. Amparo Dora Palomino Merino, al Dr. Fernando Reyes Cortés y Dr. Gibran Etcheverry Doger, por sus observaciones en este trabajo de tesis, así como los comentarios que permitieron enriquecer el presente trabajo.

Agradezco a la Vicerrectoría de Investigación y Estudio de Posgrado, por el apoyo para la presentación de este trabajo en el congreso internacional Modern Circuits and Systems Technologies, 2019, (MOCASST,2019) .

Agradezco a CONACYT por los proyectos No. 258880 (Proyecto apoyado por el fondo sectorial de Investigación para la Educación); VIEP-BUAP 2019; Plan de Trabajo CA (BUAP-CA-276), y por haberme otorgado el apoyo económico No. 629054 para estudios de posgrado.

*A Dios y a mi familia. A mi padre Gustavo y a mi madre María,
por su amor, su paciencia y sus consejos. A mi hermano Gustavo,
con quién he compartido los momentos más alegres de mi vida.
Su apoyo y confianza me han llevado hasta donde estoy ahora.*

Resumen

El presente trabajo de tesis consiste en la investigación de estrategias para la realización experimental de osciladores caóticos de orden entero y fraccional, a partir del cómputo en hardware embebido de los modelos matemáticos que describen a este tipo de sistemas. Particularmente, se ha realizado el estudio teórico del sistema de Chen y el sistema de Múltiples-Enrollamientos basado en una serie de funciones no lineales saturadas. Para la solución de sistemas de orden entero se ha descrito la formulación matemática de los métodos de Runge-Kutta de orden superior, para el caso de sistemas de orden fraccional se abordan las definiciones de cálculo fraccional en el sentido de Grunwald-Letnikov, Riemman-Liouville y de Caputo para la descripción de los métodos de Grunwald-Letnikov y de Descomposición de Adomian, las formulaciones teóricas se validan mediante simulaciones numéricas elaboradas en MATLAB. Además, se realiza el desarrollo experimental de los osciladores caóticos en una plataforma ARM SoC (BCM2837). La validación de los resultados experimentales obtenidos, se realiza mediante la construcción de mapas de Poincaré y la obtención del máximo exponente de Lyapunov de cada oscilador caótico.

Índice general

Agradecimientos	II
Resumen	IV
Lista de figuras	VII
Lista de tablas	X
1. Introducción	1
1.1. Antecedentes	2
1.2. Justificación	4
1.2.1. Objetivo general	5
1.2.2. Objetivos específicos	5
1.3. Organización del documento	6
2. Preliminares matemáticos	8
2.1. Representación en el espacio de estados de un sistema dinámico no lineal	8
2.2. Sistemas Autónomos	9
2.3. Cálculo fraccional	10
2.3.1. Derivada fraccional de Grünwald-Letnikov	11
2.3.2. Integral fraccional de Riemman-Liouville	11
2.3.3. Derivada fraccional de Caputo	12
2.4. Puntos de equilibrio	13
2.4.1. Estabilidad de puntos de equilibrio para sistemas de orden entero	13
2.4.2. Estabilidad de puntos de equilibrio para sistemas de orden frac- cional	14
2.5. Caracterización de la dinámica caótica	16

2.5.1.	Exponente Máximo de Lyapunov	16
2.5.2.	Mapas de Poincaré	18
2.6.	Métodos para el análisis numérico de sistemas de orden entero y fraccional	19
2.6.1.	Métodos de Runge-Kutta para sistemas de orden entero	20
2.6.2.	Método de Grünwald-Letnikov para sistemas de orden fraccional	21
2.6.3.	Método de Descomposición de Adomian para sistemas de orden fraccional	22
2.7.	Hardware embebido	24
2.7.1.	Filosofía de diseño RISC	25
2.7.2.	Plataformas ARM	26
2.7.3.	Raspberry Pi	26
3.	Análisis y simulación de osciladores caóticos de orden entero y fraccional	28
3.1.	Oscilador caótico de Chen	28
3.1.1.	Modelo de orden entero	28
3.1.2.	Modelo de orden fraccional	32
3.2.	Oscilador caótico de Múltiples-Enrollamientos	35
3.2.1.	Modelo de orden entero	36
3.2.2.	Modelo de orden fraccional	42
4.	Metodología para el diseño de osciladores caóticos de orden entero y fraccional	46
4.1.	Método 1: Runge-Kutta para sistemas de orden entero	46
4.2.	Método 2: Grünwald-Letnikov para sistemas de orden fraccional	48
4.2.1.	Oscilador caótico de Chen de orden fraccional basado en Grünwald-Letnikov	52
4.2.2.	Oscilador caótico de Múltiples-Enrollamientos de orden fraccional basado en Grünwald-Letnikov	53
4.3.	Método 3: Métodos de Descomposición de Adomian para sistemas de orden fraccional	55
4.3.1.	Método de Descomposición de Adomian	56
4.3.2.	Método de Descomposición de Adomian PWL	59
4.3.3.	Oscilador caótico de Chen de orden fraccional basado en ADM .	64

4.3.4. Oscilador caótico de Múltiples-Enrollamientos de orden fraccional basado en ADM PWL	67
5. Realización de osciladores caóticos de orden entero y fraccional utilizando una plataforma ARM SoC	70
5.1. Configuración de la plataforma ARM SoC	70
5.1.1. Caracterización de los convertidores Digital-Analógico	71
5.1.2. Interfaz gráfica	74
5.2. Resultados Experimentales	76
5.3. Validación del regimen caótico	80
5.3.1. Mapas de Poincaré	80
5.3.2. Exponente máximo de Lyapunov	83
6. Conclusiones y Trabajo a futuro	86
Apéndice A. Participación en Congreso y Estancia Académica	89
Evidencias de participación en Congreso	89
Evidencias de Estancia Académica	96
Bibliografía	101

Índice de figuras

2.1. Sección de Poincaré $S \in x_n$, se aprecian tres puntos de intersección P_1 , P_2 y P_3	19
2.2. Tarjeta y arquitectura de la computadora SoC Raspberry Pi 3 modelo B+.	26
3.1. Resultado de la simulación del sistema de Chen de orden entero, realizada con tiempo de simulación $T_{sim} = 90$ s, paso de integración $h = 0.008$ y condiciones iniciales $(x_1(t_0), x_2(t_0), x_3(t_0)) = (1, 0, 30)$	31
3.2. Resultado de la simulación del sistema de Chen de orden fraccional $q = 0.9$, realizada con tiempo de simulación $T_{sim} = 90$ s, paso de integración $h = 0.007$ y condiciones iniciales $(x_1(t_0), x_2(t_0), x_3(t_0)) = (1, 0, 25)$	34
3.3. Descripción de una serie de funciones saturadas para generar 4 enrollamientos.	37
3.4. Resultado de la simulación del sistema de Múltiples Enrollamientos de orden entero, realizada con tiempo de simulación $T_{sim} = 2500$ s, paso de integración $h = 0.25$ y condiciones iniciales $(x_1(t_0), x_2(t_0), x_3(t_0)) = (1, 0, 1)$	41
3.5. Resultado de la simulación del sistema de Múltiples Enrollamientos de orden fraccional $q = 0.9$, realizada con tiempo de simulación $T_{sim} = 3000$ s, paso de integración $h = 0.15$ y condiciones iniciales $(x_1(t_0), x_2(t_0), x_3(t_0)) = (1, 0, 1)$	45
4.1. Diagrama de flujo para resolver sistemas caóticos en hardware embebido por el método Runge-Kutta	47

4.2.	Enfoque numérico usual para resolver sistemas de orden fraccional con el método de Grünwald-Letnikov. Los paréntesis denotan solo las muestras que son utilizadas en el principio de memoria corta, no obstante, las muestras anteriores son almacenadas para todo el tiempo de simulación.	49
4.3.	Diagrama de flujo para resolver sistemas caóticos en hardware embebido por el método Grünwald-Letnikov.	50
4.4.	El enfoque numérico propuesto para implementar el principio de memoria corta en el método de Grünwald-Letnikov. Los paréntesis denotan las muestras utilizadas para determinar la solución en $x(t_{k+1})$	51
4.5.	Diagrama de flujo para resolver sistemas caóticos en hardware embebido por el Método de Descomposición de Adomian.	58
4.6.	Diagrama de flujo para resolver sistemas caóticos en hardware embebido por el Método de Descomposición de Adomian PWL.	63
4.7.	Resultado de la simulación del sistema de Chen de orden fraccional $q = 0.9$, realizada para $M = 10,001$ subintervalos, con $T_M = 60$ s, tamaño de paso $h = 0.006$ y condiciones iniciales $(x_1(t_0), x_2(t_0), x_3(t_0)) = (1, 0, 25)$.	66
4.8.	Resultado de la simulación del sistema de Múltiples-Enrollamientos de orden fraccional $q = 0.9$, realizada para $M = 200,001$ subintervalos, con $T_M = 2000$ s, tamaño de paso $h = 0.01$ y condiciones iniciales $(x_1(t_0), x_2(t_0), x_3(t_0)) = (1, 0, 1)$	69
5.1.	Diagrama de bloques para la validación experimental de osciladores caóticos de orden entero y fraccional.	70
5.2.	Hardware experimental utilizado para la implementación digital: 1) ARM SoC, 2) Periféricos USB para teclado y ratón, 3) Pantalla HDMI, 4) Convertidores D/A AD569 y 5) Fuente de alimentación.	71
5.3.	Numeración de los pines de entrada/salida digitales del puerto GPIO.	72
5.4.	Diagrama esquemático para la conversión D/A. Primero, se definen dos canales A, B , a partir de estos se extraen los datos en formato Q_{14} (formato binario de 14 bits) de las señales que conforman los retratos de fase x_1x_3, x_1x_2 y x_2x_3 respectivamente. Después, cada canal es conectado a un convertidor D/A AD569. Finalmente, las señales analógicas S_1 y S_2 , son acondicionadas mediante un filtro RC.	72

5.5. Respuesta típica de los convertidores D/A utilizados en el hardware experimental.	73
5.6. (a) Editor de texto para el desarrollo del programa y (b) Terminal de Linux	75
5.7. Diagrama de flujo de la implementación digital.	76
5.8. Resultados experimentales del oscilador caótico de Chen de orden entero basado en Runge-Kutta.	77
5.9. Resultados experimentales del oscilador caótico de Chen de orden fraccional basado en Grünwald-Letnikov.	78
5.10. Resultados experimentales del oscilador caótico de Chen de orden fraccional basado en ADM.	78
5.11. Resultados experimentales del oscilador caótico de Múltiples-Enrollamientos de orden entero basado en Runge-Kutta.	79
5.12. Resultados experimentales del oscilador caótico de Múltiples-Enrollamientos de orden fraccional basado Grünwald-Letnikov.	79
5.13. Resultados experimentales del oscilador caótico de Múltiples-Enrollamientos de orden fraccional basado en ADM PWL.	80
5.14. Diagrama de flujo para obtener el respectivo mapa de Poincaré de los osciladores caóticos experimentales.	81
5.15. Mapas de Poincaré de los osciladores caóticos experimentales de Chen. a) Chen de orden entero basado en Runge-Kutta, b) Chen de orden fraccional basado en Grünwald-Letnikov y c) Chen de orden fraccional basado en ADM.	83
5.16. Mapas de Poincaré de los osciladores caóticos experimentales de Múltiples-Enrollamientos. a) Múltiples-Enrollamientos de orden entero basado en Runge-Kutta, b) Múltiples-Enrollamientos de orden fraccional basado en Grünwald-Letnikov y c) Múltiples-Enrollamientos de orden fraccional basado en ADM PWL.	83

Índice de tablas

2.1. Especificaciones de la tarjeta Raspberry Pi 3 modelo B+.	27
3.1. Puntos de equilibrio del sistema de Chen de orden fraccional y sus respectivos valores propios.	33
3.2. Ubicación de los puntos de equilibrio del sistema (3.22) dentro de la función en series saturadas (3.21).	41
3.3. Membresía de los puntos de equilibrio del sistema (3.33) respecto a la función en series saturadas (3.21).	43
4.1. Resultados del principio de memoria corta en el método de Grunwald-Letnikov para el sistema de Chen de orden fraccional $q = 0.9$ a diferentes valores de L_m , con $(a, b, c) = (35, 3, 28)$, $T_{sim} = 90$ s, $h = 0.007$ y $(x(t_0), y(t_0), z(t_0)) = (1, 0, 25)$	53
4.2. Resultados del principio de memoria corta en el método de Grunwald-Letnikov para el sistema de Múltiples-Enrollamientos de orden fraccional $q = 0.9$ a diferentes valores de L_m , con $(a, b, c, d) = (2.2, 1.4, 0.6, 2.2)$, $T_{sim} = 2500$ s, $h = 0.15$ y $(x(t_0), y(t_0), z(t_0)) = (1, 0, 1)$	54
5.1. Comandos requeridos para el desarrollo de un programa en C dentro de la terminal de Linux.	75
5.2. Configuración y puntos de intersección encontrados para la generación de mapas de Poincaré de los osciladores caóticos experimentales.	82
5.3. Parámetros utilizados en el algoritmo de Rosenstein-Kantz para determinar el máximo exponente de Lyapunov.	84
5.4. Gráficas de la divergencia promedio de las trayectorias correspondientes a cada oscilador caótico.	85

Capítulo 1

Introducción

En los sistemas deterministas se tiene definido que el estado inicial determina el estado final del sistema. Si un sistema determinista presenta alta sensibilidad a sus condiciones iniciales, pequeñas variaciones tienen como resultados efectos no semejantes, es decir, una diferencia pequeña en sus estados iniciales hará que predicciones se degraden a medida que pasa el tiempo, este fenómeno no lineal es conocido como “caos”. El meteorólogo teórico estadounidense E. N. Lorenz fue el primero en reconocer el comportamiento caótico mediante el modelado matemático de un sistema para predecir condiciones meteorológicas, observó de forma empírica que existen sistemas cuya evolución temporal presentan un comportamiento quasi-aleatorio imposible de predecir a largo plazo, pero con trayectorias en el espacio de fase las cuales evolucionan desde un determinado estado inicial a una zona de atracción en la cual se encontraran en continuo movimiento y sin repetir nunca un mismo estado. Lorenz efectuó trascendentes aportes a la teoría del caos, introduciendo los conceptos de “efecto mariposa” y “atractor extraño” [1].

Los sistemas caóticos, también denominados “Osciladores caóticos”, han tenido un gran potencial en muchas disciplinas, tales como en sistemas de comunicación [2], criptografía [3], sistemas de antena y radar [4], sistemas de potencia [5], robótica [6], de ingeniería biomédica en el estudio del cerebro y del corazón humano [7, 8] por mencionar algunos casos. Además, el estudio de la dinámica caótica ha ayudado a comprender el fenómeno evolutivo de varios sistemas cuando sus parámetros asumen cierto rango de valores críticos. Por otra parte, la generación de caos tiene resultados altamente complejos, aún en sistemas de comportamientos dinámicos bien organizados, que en un principio pueden ser estables, es posible crear caos mediante la aplicación de

métodos estudiados como el acoplamiento de series de funciones no lineales saturadas (SNFS) descritas por funciones lineales por segmentos (funciones PWL) [9]. En cuanto a los sistemas autónomos es bien sabido que solo pueden exhibir caos cuando se tiene un orden mayor o igual tres. Sin embargo, se ha encontrado presencia de caos en sistemas de orden inferior, siendo estos de orden fraccional [10]. Esto quiere decir que el orden total del sistema corresponde a la suma de cada derivada fraccional. De forma general, el estudio del cálculo fraccional se ha convertido en un tema de gran interés, dado que las integrales y derivadas de orden fraccional tienen un excelente enfoque para la descripción de propiedades hereditarias o de memoria de fenómenos físicos [11]. Cabe señalar que este tipo de derivadas ofrecen una representación más compacta del sistema, dado que están definidas como una generalización de las derivadas de orden clásico, además, las derivadas de orden fraccional añaden un grado extra de libertad al sistema, y dado que sus parámetros se pueden ajustar a distintos valores de la derivada y conservar el régimen caótico, el alcance de aplicaciones basadas en caos se ha visto incrementada, por ejemplo, en aplicaciones de seguridad como el encriptamiento de datos, la derivada fraccional permite generar un espacio de claves más amplio, no obstante, las aplicaciones de sistemas caóticos tanto de orden entero como fraccional, requieren del diseño electrónico y su implementación en hardware para ser útiles en la ingeniería. Para este propósito, la literatura indica que la implementación física de osciladores caóticos se puede realizar de forma analógica donde los valores de las señales varían en un rango de tiempo continuo y de forma digital donde los valores de las señales varían en un rango de tiempo discreto [12].

1.1. Antecedentes

El “diseño analógico de osciladores caóticos” se realiza empleando circuitos analógicos elaborados a partir de amplificadores operacionales, resistencias, capacitores, inductores, diodos entre otros componentes. Esta técnica resulta en una tarea de gran complejidad debido a los efectos no lineales intrínsecos en los componentes pasivos y activos. La complejidad antes mencionada es evidente en diversas publicaciones científicas. Por ejemplo, en [13] se reportó la construcción de un oscilador caótico de orden entero donde la existencia de puntos de equilibrio depende de un único parámetro, cuyo régimen caótico fluctúa entre -0.001×10^{-3} y -0.005×10^{-3} para el caso estable e inestable respectivamente. Sin embargo, a nivel circuito esa mínima variación puede estar

relacionada con las tolerancias y los efectos de segundo orden de dispositivos electrónicos, conduciendo a que ese método sea fuertemente dependiente del diseño electrónico. En [14] se introdujo un nuevo método para la generación de caos basado en una estrategia de conmutación para el anti-control de un sistema no estable de orden entero. Aunque se realiza la implementación electrónica con amplificadores operacionales, su diseño requiere de funciones tangente hiperbólicas cuyo parámetro de control tiene el valor de 1.3919. Si este valor se modificara ligeramente por cualquier efecto no ideal de los dispositivos electrónicos, los puntos de equilibrio y los planos de conmutación se afectarían y limitarían la validez del método. Estos casos no son aislados, se han publicado diversos métodos para generar caos en sistemas caóticos de orden entero, los cuales son realizables desde un punto de vista de simulaciones computacionales. Sin embargo, a nivel experimental los efectos de segundo orden de los dispositivos electrónicos perturbarían o destruirían el comportamiento caótico. Respecto a la realización analógica de osciladores caóticos de orden fraccional la problemática es similar al caso de orden entero. No obstante, existe un problema adicional causado por el orden fraccional en las derivadas que definen al sistema dinámico [15]. Esto incrementa la complejidad para su realización experimental con dispositivos electrónicos analógicos. El proceso de diseño típico y más ampliamente utilizado para obtener una derivada de orden fraccional consiste en diseñar una cadena de resistores-capacitores en la red de retroalimentación de un amplificador operacional de ganancia inversora, resultado en una impedancia de orden fraccional, conocida como fractancia [16]. Nuevamente, las tolerancias y los efectos no ideales de los dispositivos electrónicos pasivos y activos pueden afectar no solo los valores de los parámetros del sistema sino también el orden de fraccional del sistema caótico y su consecuente comportamiento caótico. En [17] se reporta un sistema caótico sin puntos de equilibrio. En ese trabajo se propone un nuevo oscilador caótico de orden fraccional de valor 0.98. Como se observa, el comportamiento caótico de orden fraccional está comprometido a una mínima variación a nivel circuito para interferir o nulificar el comportamiento caótico.

Los problemas antes mencionados se pueden superar al adaptar propiedades teóricas a algoritmos factibles que pueden ser implementados en plataformas digitales, de esta forma el “diseño digital de osciladores caóticos” se realiza a partir de la discretización de los modelos matemáticos que representan a estos sistemas [12], para este propósito existen dos principales esquemas [18]. Por una parte, está el enfoque no-embebido orientado hacia el cómputo paralelo en FPGAs desarrolladas por algún lenguaje de descripción

de hardware como Verilog HDL, el cual describe la estructura y comportamiento del diseño digital en formato de texto. Por otra parte, se encuentra el enfoque embebido orientado hacia el cómputo secuencial en procesadores en plataformas basadas en chips ARM [19], los cuales son realizados con lenguajes de alto nivel de programación como C o Python, entonces la complejidad del diseño de los osciladores caóticos recae en la formulación de algoritmos elegantes basados en métodos numéricos:

- Para el caso clásico un sistema caótico es representado por una ODE (Ecuación Diferencial Ordinaria) de tercer orden. Las características que presentan este tipo de ecuaciones es de naturaleza no lineal y por lo tanto su solución es imposible de obtener de forma analítica. El método numérico más reconocido para estudiar este tipo de sistemas es el método de Runge-Kutta de orden superior [20], el cual presentan una gran aproximación a la solución del sistema a partir de un conjunto de valores iniciales.
- Por otro lado, el análisis numérico de derivadas de orden fraccional requiere de algoritmos que puedan representar adecuadamente las propiedades de herencia o de memoria, en trabajos como [21, 22], el método de Adams-Bashforth-Moulton (ABM) es ampliamente reconocido para para estudiar este tipo de sistemas, no obstante, es un método no apto para aplicaciones de ingeniería dado el costo computacional que requiere. En trabajos como [23, 24] se utiliza el método de Grünwald-Letnikov como alternativa a ABM para la implementación de sistemas caóticos en FPGA, reduciendo pero no evitando un alto consumo de recursos de hardware, finalmente en [25, 26] se utiliza el Método de Descomposición de Adomian (ADM) para implementaciones basados en hardware embebido, no obstante, estas implementaciones son de caso particular y no aborda la implementación de familias de sistemas caóticos basados en funciones PWL. Es por este motivo que el diseño digital de este tipo de sistemas es un área de investigación abierta.

1.2. Justificación

Actualmente el mercado electrónico tiene una fuerte expansión hacia el mundo digital, siendo el hardware embebido un concepto altamente utilizado, particularmente, las plataformas ARM se encuentran en diversos dispositivos electrónicos de consumo tales como: Smart-phones, Tablets, Laptops y otros dispositivos de desarrollo tales

como Arduinos, RaspBerrys y FPGAs. Con lo cual ha crecido un gran interés en la implementación de la dinámica caótica para el desarrollo de aplicaciones en sistemas embebidos, no obstante muchas de estas aplicaciones, no son llevadas mas allá del marco teórico dada la complejidad existente en el diseño de los osciladores caóticos. Con base en lo anterior, el presente trabajo de tesis se encuentra orientado al diseño de estrategias para la implementación embebida de sistemas caóticos de orden entero y fraccional. Con los resultados obtenidos se busca generar nuevo conocimiento científico respecto a una metodología estructurada para la realización de osciladores caóticos a través de modelos matemáticos basados en métodos numéricos. Cabe destacar que la ventaja de tener uno o varios sistemas caóticos implementados de manera digital frente a la implementación en circuitería electrónica analógica es principalmente la reducción del tamaño lo que facilita la portabilidad de los mismos.

Los objetivos planteados para el presente trabajo de tesis se describen a continuación.

1.2.1. Objetivo general

- Generar osciladores caóticos de orden entero y fraccional para su implementación electrónica usando hardware embebido.

1.2.2. Objetivos específicos

1. Determinar los sistemas caóticos de orden entero y fraccional para su estudio y caracterización de sus modelos matemáticos mediante simulaciones numéricas en MATLAB.
2. Sintetizar sistemas caóticos de orden entero, representados por modelos matemáticos, en hardware embebido, utilizando al menos uno de los siguientes: DSP, FPGA, ARM-chips.
3. Mediante la técnica desarrollada en el objetivo anterior, adaptarla para realizar la síntesis de sistemas caóticos de orden fraccional, representados por modelos matemáticos, en hardware embebido, utilizando al menos uno de los siguientes: DSP, FPGA, ARM-chips.

4. Validar los análisis anteriores mediante la realización experimental de sistemas caóticos de orden fraccional y entero para obtener el máximo exponente de Lyapunov.
5. Publicación de resultados.

1.3. Organización del documento

Este documento de tesis está organizado en cinco capítulos. En el primer capítulo se presenta una introducción general sobre el diseño de osciladores caóticos. Antecedentes, justificación y los objetivos de este trabajo.

En el segundo capítulo se introducen conceptos generales relacionados con el análisis de sistemas dinámicos no lineales. Representación en espacio de estados, análisis de estabilidad de puntos de equilibrio, criterios para la caracterización de la dinámica caótica y su desarrollo para el análisis numérico de sistemas caóticos de orden entero y fraccional. Para la solución de sistemas de orden entero se ha descrito la formulación matemática de los métodos de Runge-Kutta de orden superior, para el caso de sistemas de orden fraccional se han abordado las definiciones del cálculo fraccional en el sentido de Grünwald-Letnikov, Riemann-Liouville y de Caputo para la descripción de los métodos de Grünwald-Letnikov y Descomposición de Adomian. Finalmente se presenta una descripción del hardware utilizado para el desarrollo experimental de los osciladores caóticos.

En el tercer capítulo se realiza el análisis y simulación sistemas caóticos de orden entero y fraccional. Los estudios que se realizan en este capítulo, comprenden la evolución y la complejidad del comportamiento caótico a partir de sistemas dinámicos no lineales. Como caso de estudio se utiliza el sistema de Chen y el sistema de Múltiples Enrollamientos basado en una serie de funciones no lineales saturadas (SNFS), este último para el caso particular de cuatro enrollamientos. Los análisis antes mencionados se realizan a nivel simulación utilizando el software de programación MATLAB.

En el cuarto capítulo se presenta la descripción de la metodología basada en modelos matemáticos para el diseño de osciladores caóticos de orden entero y fraccional. Esta metodología comprende la implementación de algoritmos eficientes basados en los métodos de Runge-Kutta para el caso clásico y los métodos de Grünwald-Letnikov y de Descomposición de Adomian para el caso de orden fraccional. Además, se desarrolla

una extensión al método tradicional de Adomian para la solución de sistemas basados en funciones lineales por segmentos (PWL).

En el quinto capítulo se lleva a cabo la realización experimental de estos sistemas en una plataforma ARM SoC (BCM2837), utilizando el lenguaje de programación C para la implementación de los algoritmos formulados. La validación de los resultados experimentales se realiza mediante la construcción de mapas de Poncaire y la obtención del máximo exponente de Lyapunov de cada oscilador caótico.

Finalmente se presentan las conclusiones de este trabajo. Parte de los resultados obtenidos durante el desarrollo de esta Tesis, se han presentado en un congreso internacional y han servido para el desarrollo de un proyecto de investigación en el extranjero.

Capítulo 2

Preliminares matemáticos

2.1. Representación en el espacio de estados de un sistema dinámico no lineal

Un sistema dinámico se caracteriza por un conjunto de variables relacionadas que pueden cambiar con el tiempo, para hacer un progreso cuantitativo en la comprensión del comportamiento de este sistema, se necesita un modelo matemático. Tal modelo puede formularse de muchas maneras, pero su característica esencial es permitir predecir el comportamiento futuro del sistema dada su condición inicial y el conocimiento de las fuerzas externas que lo afectan. La estructura matemática más naturalmente adaptada para este propósito es la representación en el espacio de estados [27]. Esta representación consiste en un conjunto de ecuaciones diferenciales de primer orden las cuales describen la evolución de las variables del sistema, estas se conocen como las variables de estado y sus condiciones iniciales para un tiempo fijo deben contener información suficiente para predecir la evolución del sistema dado que las influencias externas (o variables de entrada) que actúan sobre él son conocidas. Se define la siguiente representación en el espacio de estados de un sistema no lineal como:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}, \mathbf{u}, t),\end{aligned}\tag{2.1}$$

donde el punto denota la derivada con respecto al tiempo (t), las funciones $\mathbf{f}$ y $\mathbf{h}$ son en general, no lineales. Para facilitar la notación, es común recopilar las variables de

estado en un vector $\mathbf{x}(t) \in \mathbb{R}^n$. Las funciones no lineales pueden surgir en un modelo dinámico ya sea porque son intrínsecas a la naturaleza del sistema o porque en un caso como un sistema de control han sido deliberadamente introducidos por el diseñador para un propósito específico [28]. Primero, existen funciones analíticas simples como funciones sinusoides y exponenciales de una sola variable, o productos de diferentes variables. Una característica importante de estas funciones es que son lo suficiente suaves como para poseer convergentes expansiones de Taylor en todos los puntos y por lo tanto es posible linealizarlas [27]. Un tipo de función no lineal de uso frecuente en modelado de sistemas es la aproximación PWL [29, 30] que consiste en un conjunto de relaciones lineales válidas en diferentes regiones. Tienen la ventaja de pasar de ecuaciones dinámicas no lineales a ecuaciones lineales (por lo tanto, solucionables) en cualquier región en particular y las soluciones para diferentes regiones se pueden unir en los límites. Aunque el objetivo de esta tesis se refiere principalmente a fenómenos no lineales, es apropiado revisar la representación de estados para el caso de sistemas lineales, dado que la linealización es ampliamente aplicable para resolver problemas de sistemas no lineales [31]. Se define la siguiente representación de un sistema lineal como:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t),\end{aligned}\tag{2.2}$$

donde el punto denota la derivada con respecto al tiempo (t), $\mathbf{x}(t) \in \mathbb{R}^n$ es el vector de estados, $\mathbf{A}(t) \in \mathbb{R}^{n \times n}$ es la matriz del sistema, $\mathbf{B}(t) \in \mathbb{R}^{n \times m}$ es el vector de entrada, $\mathbf{C}(t) \in \mathbb{R}^{p \times n}$ es el vector de salida y $\mathbf{D}(t) \in \mathbb{R}^{p \times m}$ es la matriz de transmisión directa (en la mayoría de sistemas es nula), note que al ser un sistema no autónomo las matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ y $\mathbf{D}$ se encuentran en función del tiempo [32].

2.2. Sistemas Autónomos

Aunque las ecuaciones de un modelo dinámico, en general, dependen del tiempo ya sea de manera explícita, a través de la función de entrada $u(t)$, o ambas, gran parte de la teoría de sistemas no lineales se refiere a casos en los que no existe dependencia del tiempo [33]. Estos sistemas son autónomos y surgen de manera bastante natural en la práctica, por ejemplo, cuando el vector de entrada se mantiene fijo. En cualquier

caso, la representación en espacio de estados de la ecuación diferencial de un sistema no lineal toma la siguiente forma [27]:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \hat{\mathbf{u}}), \quad (2.3)$$

donde $\hat{\mathbf{u}}$ es un vector constante, Por lo tanto, los puntos de equilibrio en el espacio de estados están determinados por $\mathbf{f}(\mathbf{x}, \hat{\mathbf{u}}) = 0$. Asumiendo que $\mathbf{f}(\mathbf{x}, \hat{\mathbf{u}})$ satisface la condición de Lipschitz, la ecuación diferencial tendrá una única solución $x(t) \in \mathbb{R}^n$ para cualquier condición inicial $x(0)$ [27]. La trayectoria trazada en el espacio de fase por $x(t)$ se llama trayectoria del sistema y debido a la propiedad de unicidad [34], habrá una y solo una trayectoria que pase por cualquier punto dado. Si se suprime la dependencia de $\hat{\mathbf{u}}$, la representación en espacio de estados para un sistema no lineal autónomo puede escribirse simplemente:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \quad (2.4)$$

y el conjunto de todas las trayectorias de esta ecuación proporciona una representación geométrica completa del comportamiento dinámico no lineal del sistema bajo ciertas condiciones iniciales. Para el caso de un sistema lineal autónomo o invariante en el tiempo la representación en espacio de estados se escribe de la siguiente manera:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t). \end{aligned} \quad (2.5)$$

Por lo general suele omitirse la variable independiente de tiempo t de la representación (2.5) por ser implícita.

2.3. Cálculo fraccional

El cálculo fraccional se refiere a la generalización de derivadas e integrales de orden arbitrario. Este es considerado un tema tanto antiguo como novedoso dado que fue discutido por Leibniz y L'Hospital en 1665, sin embargo, es recientemente que las aplicaciones del cálculo fraccional en diversos campos de la ciencia e ingeniería se han convertido en un foco de gran interés [35]. Como se explica en [36], la definición de

derivada e integral fraccional no es única e inicia con responder el significado de $\frac{d^q}{dt^q}f(t)$ cuando $0 < q < 1$, es necesario remarcar que la definición puede ser generalizada para incluir cualquier $q \in \mathbb{R}_+$. A continuación se presentan las definiciones y métodos utilizados para analizar sistemas caóticos de orden fraccional.

2.3.1. Derivada fraccional de Grünwald-Letnikov

De acuerdo con Grünwald-Letnikov [36], la q -ésima derivada fraccional para una función continua $f(t)$ se define de la siguiente manera

$$D_t^q f(t) = \lim_{h \rightarrow 0} \frac{1}{h^q} \sum_{j=0}^{\infty} (-1)^j \binom{q}{j} f(t - jh), \quad (2.6)$$

donde D_t^q es el operador derivativo de Grünwald-Letnikov y $\binom{q}{j}$ son los coeficientes binomiales calculados de la siguiente forma:

$$\binom{q}{0} = 1, \quad \binom{q}{j} = \frac{\Gamma(q+1)}{\Gamma(j+1)\Gamma(q-j+1)}, j = 1, 2, \dots, \infty. \quad (2.7)$$

Aquí se debe mencionar la función más importante utilizada en el cálculo fraccional, conocida como la función Gamma de Euler $\Gamma(\cdot)$, la cual es definida como la generalización del factorial $(q-1)!$, $q \in \mathbb{R}_+$:

$$\Gamma(q) = \int_0^{\infty} t^{q-1} e^{-t} dt. \quad (2.8)$$

2.3.2. Integral fraccional de Riemman-Liouville

De acuerdo con el enfoque de Riemman-Liouville del cálculo fraccional, el concepto de una integral fraccional de orden $q > 0$, se encuentra dado por la siguiente integral de tipo convolución [37]:

$$J_{t_0}^q f(t) := \frac{1}{\Gamma(q)} \int_{t_0}^t (t - \tau)^{q-1} f(\tau) d\tau, \quad t \in [t_0, t_f], \quad q \in \mathbb{R}^+, \quad (2.9)$$

donde $J_{t_0}^q$ es el operador integral fraccional de Riemman-Liouville de orden $q > 0$, el cual existe para casi todos los intervalos $t \in [t_0, t_1]$, que se define como el operador identidad $J_{t_0}^0 = I$, cuando $q = 0$. A continuación se describen las propiedades fundamentales de

este operador:

- (a) *Commutatividad y aditividad*

$$J_{t_0}^q J_{t_0}^\alpha f(t) = J_{t_0}^{q+\alpha} f(t) \iff J_{t_0}^\alpha f(t) J_{t_0}^q f(t) = J_{t_0}^{\alpha+q} f(t); \quad (2.10)$$

- (b) *Dada una función $f(t) = t^\gamma$, $\gamma > -1$*

$$J_{t_0}^q t^\gamma = \frac{\Gamma(\gamma + 1)}{\Gamma(\gamma + 1 + q)} (t - t_0)^{\gamma+q}; \quad (2.11)$$

- (c) *Dada una constante $C \in \mathbb{R}$*

$$J_{t_0}^q C = \frac{C}{\Gamma(q + 1)} (t - t_0)^q. \quad (2.12)$$

2.3.3. Derivada fraccional de Caputo

Una definición de derivada fraccional es propuesta por Caputo [37], dado por el operador derivativo fraccional $*D_{t_0}^q = J_{t_0}^{m-q} D_{t_0}^m f(t)$ con $m \in \mathbb{N}^+$, $q \in \mathbb{R}^+$, $m - 1 < q \leq m$, este operador está definido de la siguiente manera:

$$*D_{t_0}^q f(t) := \begin{cases} \frac{1}{\Gamma(m-q)} \int_{t_0}^t \frac{f^{(m)}(\tau)}{(t-\tau)^{q+1-m}} d\tau & m - 1 < q \leq m, \\ \frac{d^m}{dt^m} f(t) & q = m. \end{cases} \quad (2.13)$$

Al aplicar el operador $*D_{t_0}^q$ al operador integral fraccional $J_{t_0}^q$, se tiene el operador identidad $*D_{t_0}^q J_{t_0}^q = I$, lo cual implica que $*D_{t_0}^q J_{t_0}^q f(t) = f(t)$. No obstante dicha operación no es conmutativa, es decir $*D_{t_0}^q J_{t_0}^q f(t) \neq *J_{t_0}^q D_{t_0}^q f(t)$; Observe que al expandir $*J_{t_0}^q D_{t_0}^q f(t)$ se obtiene la siguiente expresión:

$$\begin{aligned} J_{t_0}^q *D_{t_0}^q f(t) &= J_{t_0}^q (J_{t_0}^{m-q} D_{t_0}^m f(t)), \\ &= J_{t_0}^m D_{t_0}^m f(t), \\ &= f(t) - \sum_{k=0}^{m-1} f^{(k)}(t_0^+) \frac{(t - t_0)^k}{k!}. \end{aligned} \quad (2.14)$$

2.4. Puntos de equilibrio

Los puntos de equilibrio de un sistema autónomo $D_t^q \mathbf{x} = \mathbf{f}(\mathbf{x})$, $\mathbf{x}(t) \in \mathbb{R}^n$ con $q = 1$ y $0 < q < 1$ para el caso clásico y fraccional, respectivamente, también se conocen como puntos singulares cuando $\mathbf{f}(\bar{\mathbf{x}}) = \mathbf{0}$. Denotados como $\bar{\mathbf{x}} \in \mathbb{R}^n$ parecen violar la regla general de que sólo una trayectoria puede pasar por un punto determinado [28]. En realidad, la violación es solo aparente, ya que las trayectorias que se encuentran en un punto singular realmente no pasan a través de él, sino solo se acercan o se alejan de él asintóticamente [34].

2.4.1. Estabilidad de puntos de equilibrio para sistemas de orden entero

La linealización [38] responde al objetivo de “aproximar localmente” una función no lineal por una lineal, siempre más sencilla de manejar. Considere las funciones $f_i(x_1, x_2, \dots, x_n)$, cuyas derivadas $\frac{\partial f_i}{\partial x_j}$, con $j = 1, 2, \dots, n$ para algún i son continuas en el conjunto abierto $\Omega \in \mathbb{R}^n$, por tanto, se garantiza que las funciones f_i son diferenciables en todo punto de Ω .

Sea entonces $\bar{\mathbf{x}} = [\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n]^T$ un punto de equilibrio para sistema no lineal dado por $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$. Se llama linealización o aproximación lineal a la matriz jacobiana $\mathbf{J}(\mathbf{x}) \in \mathbb{R}^{n \times n}$ evaluada en $\bar{\mathbf{x}}$, es decir:

$$\mathbf{J}(\bar{\mathbf{x}}) = \begin{bmatrix} \frac{\partial}{\partial x_1} f_1(x_1, x_2, \dots, x_n) & \frac{\partial}{\partial x_2} f_1(x_1, x_2, \dots, x_n) & \cdots & \frac{\partial}{\partial x_n} f_1(x_1, x_2, \dots, x_n) \\ \frac{\partial}{\partial x_1} f_2(x_1, x_2, \dots, x_n) & \frac{\partial}{\partial x_2} f_2(x_1, x_2, \dots, x_n) & \cdots & \frac{\partial}{\partial x_n} f_2(x_1, x_2, \dots, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial x_1} f_n(x_1, x_2, \dots, x_n) & \frac{\partial}{\partial x_2} f_n(x_1, x_2, \dots, x_n) & \cdots & \frac{\partial}{\partial x_n} f_n(x_1, x_2, \dots, x_n) \end{bmatrix}, \quad (2.15)$$

donde

$$\left. \frac{\partial f_i}{\partial x_j} \right|_{(x_1, x_2, \dots, x_n) = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)}.$$

El objetivo de la linealización es que las propiedades de estabilidad para un punto de equilibrio $\bar{\mathbf{x}}$ de un sistema no lineal $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ sean las mismas que de un sistema lineal $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t)$. Si $\bar{\mathbf{x}}$ es un punto de equilibrio hiperbólico, es decir, tiene valores propios con parte real diferente de cero $Re(\lambda_i) \neq 0$, se determina por el **teorema de**

Hartman-Grobman, que el comportamiento de las soluciones $\mathbf{x}(t) \in \mathbb{R}^n$ cercanas al punto de equilibrio $\bar{\mathbf{x}}$, es topológicamente conjugado (existe un homeomorfismo, es decir, una función que tiene inversa y ambas continuas definida en un entorno abierto U del origen), a las soluciones del sistema linealizado. Esto implica que la estabilidad del punto de equilibrio $\bar{\mathbf{x}}$ es equivalente respecto al comportamiento de las trayectorias en un entorno cercano del mismo.

Sean $\lambda_1, \lambda_2, \dots, \lambda_n$ los valores propios de $\mathbf{J}(\bar{\mathbf{x}})$:

- 1) Si $\operatorname{Re}(\lambda_i) < 0 \ \forall i = 1, 2, \dots, n$ entonces el equilibrio $\bar{\mathbf{x}}$ es asintóticamente estable en forma local. Es tipo atractor local (sumidero).
- 2) Si $\operatorname{Re}(\lambda_i) > 0 \ \forall i = 1, 2, \dots, n$ entonces el equilibrio $\bar{\mathbf{x}}$ es inestable. Es tipo repulsor local (fuente).
- 3) Si $\operatorname{Re}(\lambda_i) > 0$ para algún j , y $\operatorname{Re}(\lambda_k) < 0$ para algún $k \neq j$ entonces el equilibrio $\bar{\mathbf{x}}$ es inestable. Es un punto tipo silla.
- 4) si $\operatorname{Re}(\lambda_i) = 0$ para algún j , entonces no se puede concluir nada sobre la estabilidad de $\bar{\mathbf{x}}$.

Para tener obtener una formulación teórica al atractor caótico, es de interés identificar los puntos de equilibrio $\mathbf{x}$ tipo silla de índice uno y dos, por simplicidad, sea $n = 3$:

- Un punto de equilibrio $\bar{\mathbf{x}}$ tiene estabilidad de tipo silla de índice uno si le corresponden valores propios que satisfacen $\lambda_1 = \alpha_1 > 0$ y $\lambda_{2,3} = \alpha \pm \beta i$ con $\alpha < 0$ y $\beta \in \mathbb{R}$.
- Un punto de equilibrio $\bar{\mathbf{x}}$ tiene estabilidad de tipo silla de índice dos si le corresponden valores propios que satisfacen $\lambda_1 = \alpha < 0$ y $\lambda_{2,3} = \alpha \pm \beta i$ con $\alpha > 0$ y $\beta \neq 0$.

Donde el número de puntos de equilibrio tipo silla de índice dos indican el número de enrollamientos presentes en el atractor caótico, los cuales se encuentran unidos por los puntos de equilibrio tipo silla de índice uno.

2.4.2. Estabilidad de puntos de equilibrio para sistemas de orden fraccional

El análisis de estabilidad para los puntos de equilibrio de un sistema de orden fraccional se divide cuando el sistema es de orden conmensurado y cuando es de orden

incommensurado. Considere el siguiente sistema de orden fraccional:

$$\begin{aligned} D_t^{q_1} x_1 &= f_1(x_1, x_2, \dots, x_n), \\ D_t^{q_2} x_2 &= f_2(x_1, x_2, \dots, x_n), \\ &\vdots \\ D_t^{q_n} x_n &= f_n(x_1, x_2, \dots, x_n), \end{aligned} \quad (2.16)$$

donde $D_t^{q_i}$ expresa la derivada de orden fraccional, con $i = 1, 2, \dots, n$.

Para el caso de orden conmensurado $q_1 = q_2 = \dots = q_n$, de acuerdo con el teorema de **Estabilidad para orden fraccional conmensurado** [36], el punto de equilibrio $[\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n]^T$ del sistema (2.16) es asintóticamente estable si todos los valores propios λ_i , ($i = 1, 2, \dots, n$), de la matriz jacobiana de (2.16) evaluada en el punto de equilibrio $\mathbf{J}(\bar{\mathbf{x}})$, satisface la condición:

$$|\arctan(\text{eig}(\mathbf{J}(\bar{\mathbf{x}})))| = |\arctan(\lambda_i)| > q \frac{\pi}{2}, \quad i = 1, 2, \dots, n. \quad (2.17)$$

Para el caso de orden incommensurado $q_1 \neq q_2 \neq \dots \neq q_n$, de acuerdo con el teorema de **Estabilidad para orden fraccional incommensurado** [36], se define m al mínimo común múltiplo del denominador u_i de q_i , donde $q_i = v_i/u_i$, con $v_i, u_i \in \mathbb{Z}^+$ para $i = 1, 2, \dots, n$ y se define $\gamma = 1/m$. Entonces el punto de equilibrio $[\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n]^T$ es asintóticamente estable si:

$$|\arctan(\lambda_i)| > \gamma \frac{\pi}{2}, \quad (2.18)$$

para todas las raíces λ de la siguiente ecuación:

$$\det(\text{diag}[\lambda^{mq_1} \lambda^{mq_2} \dots \lambda^{mq_n}] - \mathbf{J}(\bar{\mathbf{x}})) = 0. \quad (2.19)$$

Una condición necesaria para que un sistema de orden fraccional presente un comportamiento caótico, es que al menos un valor propio λ se encuentre dentro de la región inestable. Asumiendo que un atractor caótico de tres dimensiones tiene solo tres puntos de equilibrio, si el sistema tiene un atractor de doble enrollamiento, este tiene dos puntos de equilibrio tipo silla de índice dos rodeados por estos enrollamientos, más un punto de equilibrio tipo silla adicional de índice uno. Suponiendo que un sistema tiene los valores propios correspondientes a un enrollamiento $\lambda_{2,3} = \alpha \pm i\beta$. La condi-

ción necesaria para que el sistema presente un atractor de doble enrollamiento, es que los valores propios $\lambda_{2,3}$ permanezcan en la region inestable. La condición minima para sistemas de orden fraccional conmensurado se expresa en la siguiente ecuación [39]:

$$q > \frac{2}{\pi} \min \left(\arctan \left(\frac{|\beta_j|}{\alpha_j} \right) \right), j = 1, 2. \quad (2.20)$$

En otras palabras, cuando $q = \frac{2}{\pi} \min \left(\arctan \left(\frac{|\beta_j|}{\alpha_j} \right) \right) + \delta$ tal que $q \leq 1$, existe un intervalo donde el sistema (2.16) exhibe un comportamiento caótico.

2.5. Caracterización de la dinámica caótica

El caos se refiere a un tipo de comportamiento dinámico no lineal el cual posee características muy especiales tales como: extrema sensibilidad a pequeñas variaciones de las condiciones iniciales y de sus parámetros, trayectorias limitadas en el espacio de fase pero con un exponente máximo de Lyapunov positivo. De acuerdo con [40] existen 3 identificadores básicos para que exista el caos determinista. Un sistema dinámico no lineal debe cumplir con los siguientes puntos:

- 1) Tener un conjunto denso de órbitas periódicas, La densidad en órbitas periódicas implica que cualquier trayectoria periódica de la órbita visita un barrio arbitrariamente pequeño de una no periódica.
- 2) Ser transitivo, la transitividad se relaciona con la existencia de los puntos a, b para lo cual se puede encontrar un tercer punto c que está arbitrariamente cerca de a y cuya órbita pasa arbitrariamente cerca de b .
- 3) Sensible a las condiciones iniciales. Finalmente, la sensibilidad a las condiciones iniciales es la propiedad de qué condiciones similares den lugar a órbitas que finalmente se separan por una cantidad finita.

2.5.1. Exponente Máximo de Lyapunov

El estudio de las trayectorias de un sistema dinámico es un diagnóstico fundamental para determinar si este tiene un comportamiento caótico. Considere dos trayectorias que inician cercanas entre sí, en $x(t_0) = f(x(t_0))$ y $x(t_0) + \delta x(t_0) = f(x(t_0) + \delta x(t_0))$, si la evolución de las trayectorias convergen hacia una misma órbita, el atractor es un

punto fijo o un ciclo límite. Sin embargo, si estas trayectorias divergen y más aún, si se separan de forma exponencial en un tiempo finito, entonces el atractor es caótico, un indicador general que se utiliza para determinar y cuantificar este comportamiento, es el exponente máximo de Lyapunov (LLE) [41]:

$$\lambda_{max} = \lim_{t \rightarrow \infty} \frac{1}{t} \ln \left| \frac{\delta x(t)}{\delta x(t_0)} \right|. \quad (2.21)$$

En general un sistema n -dimensional tiene n exponentes de Lyapunov. No obstante para determinar si una serie de tiempo presenta una régimen caótico, se procede a calcular el exponente máximo de Lyapunov, cuando $\lambda_{max} > 0$ entonces la serie de tiempo analizada presenta un comportamiento caótico.

Obtener el LLE es una tarea compleja independiente del orden del sistema. Por tal motivo se recurre al uso de métodos numéricos. Para este caso se utiliza el algoritmo de Rosenstein-Kantz [42], el cual reconstruye una aproximación de la dinámica del sistema a partir de un conjunto de datos proporcionado de la siguiente manera:

Sea $\mathbf{X}$ una serie de tiempo para alguna variable del sistema $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))$, asumiendo que $\mathbf{X} = [X_1, X_2, \dots, X_N]^T$, la reconstrucción del espacio de fase se encuentra dada por la siguiente expresión:

$$\mathbf{X} = [X_i, X_{i+j}, \dots, X_{i+(m-1)j}]^T, \quad (2.22)$$

donde $i = 1, 2, \dots, M$, $M = N - (m-1)j$, j es el retardo de construcción de la trayectoria, m es la dimensión embebida del sistema determinada por el número de espacios de fase o variables de estado del sistema.

En la reconstrucción de la dinámica, el algoritmo localiza el vecino más cercano de cada punto de la trayectoria

$$d_j(0) = \min_X |X_{i+j} - X_i|, \quad (2.23)$$

donde $d_j(0)$ es la distancia inicial del primer punto con su vecino más cercano. Esto es encontrar el punto más cercano de un punto de referencia inicial, como restricción adicional se supone que este punto tiene una separación temporal superior al recíproco

del espectro de frecuencia:

$$d_j(0) > \left[\text{Freqmax} \left(\left| \sum_n^{N-1} x(n) e^{-j2\pi k \frac{n}{N}} \right| \right) \right]^{-1}. \quad (2.24)$$

Esto permite considerar cada parte de las vecindades como una condición inicial para diferentes trayectorias, como resultado se tiene una construcción de la divergencia promedio desde el punto inicial:

$$d_j(t_k) = \frac{1}{k} \sum_{i=1}^k \ln \left(|X_{i+j} - X_i| \right), \quad k = i + (m-1)j. \quad (2.25)$$

Finalmente, se gráfica $d_j(t_k)$ contra el número de puntos k , para la cual existe una región cuya pendiente es igual al valor teórico λ_{max} .

2.5.2. Mapas de Poincaré

Los mapas de Poincaré permiten que representación de un espacio n dimensional sea reducido a una representación de $(n-1)$ dimensiones. Esto permite identificar la periodicidad (si existe) de una trayectoria o, cuando su comportamiento completo en el espacio de fase es muy complicado (e.g $n > 3$). Tales mapas son valiosos para un análisis más detallado, Comportamiento caóticos y de otro tipo se pueden distinguir visualmente entre sí dado que preserva las propiedades de estabilidad del sistema [43]. Los pasos para la construcción de un mapa de Poincaré se explican a continuación [44]. Considere el siguiente sistema autónomo

$$\begin{aligned} D_t^q x_1 &= f_1(x_1, x_2, \dots, x_n), \\ D_t^q x_2 &= f_2(x_1, x_2, \dots, x_n), \\ &\vdots \\ D_t^q x_n &= f_n(x_1, x_2, \dots, x_n), \end{aligned} \quad (2.26)$$

con $q = 1$ y $0 < q < 1$ para el caso clásico y fraccional,

- Primero, se define una sección de Poincaré S , (tal como se muestra en la figura 2.1). Sin pérdida de generalidad, se considera $S \in x_n$. De esta forma, se capturan los puntos de intersección de la trayectoria $[x_1, x_2, \dots, x_{n-1}]^T$, cada vez que x_n

pasa por S .

- Por simplicidad, las trayectorias asociadas a esta intersección, se almacenan en los vectores $\mathbf{G}_1, \mathbf{G}_2, \dots, \mathbf{G}_{n-1} \in \mathbb{R}^k$ respectivamente, donde la dimensión para cada vector, es determinado por la k -ésima intersección de la trayectoria $[x_1, x_2, \dots, x_{n-1}]^T$ con S .
- Finalmente, al graficar en el espacio de fase los vectores obtenidos, se forma un plano denominado Mapa de Poincaré $\Sigma(\mathbf{G}_1, \mathbf{G}_2, \dots, \mathbf{G}_{n-1})$.

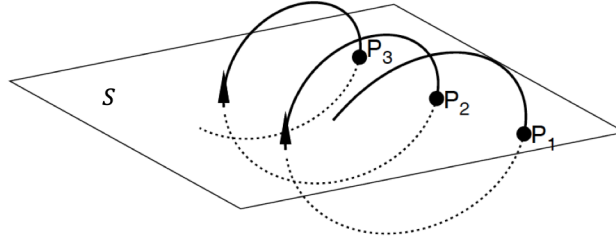


Figura 2.1: Sección de Poincaré $S \in x_n$, se aprecian tres puntos de intersección P_1 , P_2 y P_3 .

Cabe recalcar, que el valor para la sección de Poincaré no puede ser determinado de forma analítica para casi cualquier flujo de interés físico, ya que requiere de cierto conocimiento de las estructuras geométricas del espacio de fase del sistema.

2.6. Métodos para el análisis numérico de sistemas de orden entero y fraccional

Por lo general, las ecuaciones diferenciales que describen un sistema dinámico no lineal no pueden resolverse de forma analítica, de modo que, para poder estudiarlos es necesario utilizar métodos de integración numérica ó algoritmos de aproximación numérica. Estos, apuntan a la reconstrucción de trayectorias a partir de un valor inicial conocido y su planteamiento matemático implica una discretización del sistema dinámico, no obstante, vale la pena mencionar que cualquier método o algoritmo implementado en una plataforma digital tiene resultados físicos implícitamente discretos.

2.6.1. Métodos de Runge-Kutta para sistemas de orden entero

Los métodos de Runge-Kutta son reconocidos como uno de los más importantes para el estudio de sistemas no lineales de orden entero (2.4). Dichos métodos se basan en aproximar la solución a través de una expansión en series de Taylor [45].

$$x(t_{k+1}) = x(t_k) + hf(t_k, x(t_k)) + \frac{h^2}{2!}f'(t_k, x(t_k)) + \frac{h^3}{3!}f''(t_k, x(t_k)) + \dots + \frac{h^p}{p!}f^{p-1}(t_k, x(t_k)), \quad (2.27)$$

para el p -ésimo término de interés, t_k es el tiempo discreto y $h = (t_{k+1} - t_k)$ es el paso de integración, el cual especifica la distancia entre subintervalos o iteraciones $(x(t_0), x(t_1), x(t_2), \dots, x(t_N))$ para una trayectoria de interés, donde

$$N = \frac{x(t_N) - x(t_0)}{h} + 1. \quad (2.28)$$

Dentro de esta familia, el método más simple es el método de Euler ó Runge-Kutta de primer orden, este utiliza una serie de expansión de Taylor de primer orden, y se encuentra dado de la siguiente manera:

$$\begin{aligned} k_1 &= f(t_k, x(t_k)), \\ x(t_{k+1}) &= x(t_k) + hk_1. \end{aligned} \quad (2.29)$$

En general, el error global de los métodos de Runge-Kutta (error acumulado tras N iteraciones) es proporcional a h^{p-1} . Para el método de primer orden, esto implica una correlación lineal del error con el paso de integración, por tal motivo su convergencia requiere de valores muy pequeños de h . Como se observa en la expresión (2.28) esto implica un mayor número de iteraciones para lograr una aproximación aceptable en la solución del sistema. Es por tal motivo que en la práctica se utilizan métodos de Runge-Kutta de orden superior, tal es el caso del el método de Runge-Kutta cuarto orden, el cual es reconocido como una valiosa herramienta de cálculo por la buena aproximación que produce [46].

El método de Runge-Kutta de cuarto orden se encuentra dado de la siguiente ma-

nera:

$$\begin{aligned}
 k_1 &= hf(t_k, x_{t_k}), \\
 k_2 &= hf\left(t_k + \frac{1}{2}h, x_{t_k} + \frac{1}{2}k_1\right), \\
 k_3 &= hf\left(t_k + \frac{1}{2}h, x_{t_k} + \frac{1}{2}k_2\right), \\
 k_4 &= hf(t_k + h, x_{t_k} + k_3), \\
 x(t_{k+1}) &= x(t_k) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4).
 \end{aligned} \tag{2.30}$$

Dado que este método utiliza una expansión de Taylor de cuarto orden ($p = 5$), el error global es proporcional a h^4 .

2.6.2. Método de Grünwald-Letnikov para sistemas de orden fraccional

Como se ha mencionado anteriormente, las derivadas de orden fraccional son consideradas como una generalización de las de orden entero, esto se puede observar claramente en la definición presentada por Caputo (2.13). No obstante, la definición presentada por Grünwald-Letnikov (2.6) resulta una opción más conveniente para propósitos de análisis numérico [47]. En este sentido, la solución numérica para un sistema de orden fraccional

$$D_{t_k}^q = f(t_k, x(t_k)), \tag{2.31}$$

se obtiene de la siguiente manera:

$$x(t_{k+1}) = f(t_k, x(t_k))h^q - \sum_{j=1}^k C_j^q x(t_{k-j}), \tag{2.32}$$

donde los coeficientes binomiales (2.7) expresados en la forma C_j^q , son calculados de la siguiente forma recursiva

$$C_0^q = 1, \quad C_j^q = \left(1 - \frac{1+q}{j}\right) C_{j-1}^q, \quad j = 1, 2, \dots, k. \tag{2.33}$$

Cabe señalar que la propiedad de herencia en la derivada de Grünwald-Letnikov es representada mediante el operador de suma, por lo tanto, la expresión (2.32) requiere

el conocimiento de toda la historia pasada o memoria, esto significa que esta derivada fraccional a diferencia de las derivadas estándar, ya no es local sino global. Para propósitos de análisis numérico utilizar la memoria completa permite mayor aproximación al valor real de la derivada, no obstante, para aplicaciones practicas esto implica un gran consumo de recursos computacionales. Para solventar este problema, el índice superior de la suma necesita ser modificado por $k = v$, donde

$$v = \begin{cases} 1 & k < \frac{L_m}{h}, \\ \frac{L_m}{h} & k \geq \frac{L_m}{h}. \end{cases}$$

Esto es conocido como el “principio de memoria corta” en el método de Grünwald-Letnikov, donde L_m es la longitud de memoria propuesto. Cuando se utiliza esta simplificación, el método aproxima la solución $x(t_{k+1})$ utilizando solo los valores que se encuentran dentro de un intervalo fijo de tiempo $[t_k - \frac{L_m}{h}, t_N]$.

Es evidente que existe una compensación al utilizar la técnica anterior, es decir, la demanda de recursos computacionales disminuye pero también disminuye la aproximación al valor de la solución. Especificando la longitud de memoria L_m y si $|f(t_k)| \leq M$, $\forall a \leq t_k \leq b$, es posible definir y estimar el error $\Delta(t)$ para un $\varepsilon > \Delta(t)$, de la siguiente forma [48]:

$$L_m \geq \left(\frac{M}{\varepsilon |\Gamma(1-q)|} \right)^{1/q}. \quad (2.34)$$

2.6.3. Método de Descomposición de Adomian para sistemas de orden fraccional

El Método de Descomposición de Adomian (ADM) es considerado como un método analítico para resolver ecuaciones diferenciales de orden entero y fraccional [49]. La idea fundamental de este método es realizar una expansión en los “denominados polinomios de Adomian”, de las no linealidades que se presentan en un sistema dinámico.

De acuerdo con la definición de derivada fraccional de Caputo (2.13), considere el

siguiente sistema de orden fraccional:

$$\begin{aligned} {}^*D_{t_0}^q x_1(t) &= f_1(x_1, x_2, \dots, x_n), \\ {}^*D_{t_0}^q x_2(t) &= f_2(x_1, x_2, \dots, x_n), \\ &\vdots \\ {}^*D_{t_0}^q x_n(t) &= f_n(x_1, x_2, \dots, x_n), \end{aligned} \quad (2.35)$$

donde $n \in \mathbb{N}^+$. El sistema anterior es dividido en tres partes como se muestra en la siguiente expresión [50]:

$${}^*D_{t_0}^q \mathbf{x}(t) = L\mathbf{x}(t) + N\mathbf{x}(t) + \mathbf{g}(t), \quad (2.36)$$

donde ${}^*D_{t_0}^q$ es el operador derivativo de Caputo, $\mathbf{x}(t) \in \mathbb{R}^n$ es el vector de variables de estado, L y N representan el operador lineal y no lineal respectivamente y $\mathbf{g}(t) \in \mathbb{R}^n$ es una función analítica conocida. Al aplicar el operador $J_{t_0}^q$ a ambos lados de (2.36), se obtiene la siguiente expresión:

$$\mathbf{x}(t) = \sum_{k=0}^{m-1} \mathbf{f}^{(k)}(t_0^+) \frac{(t_1 - t_0)^k}{k!} + J_{t_0}^q L\mathbf{x}(t) + J_{t_0}^q N\mathbf{x}(t) + J_{t_0}^q \mathbf{g}(t). \quad (2.37)$$

Por simplicidad denotamos $\boldsymbol{\eta} = \sum_{k=0}^{m-1} \mathbf{f}^{(k)}(t_0^+) \frac{(t_1 - t_0)^k}{k!}$, considerando $m = 1$, el vector $\boldsymbol{\eta}$ se define como el vector de condiciones iniciales $\mathbf{f}^{(0)}(t_0^+) = \mathbf{x}^{(0)}(t_0)$. Por otra parte, el vector no lineal del sistema $N\mathbf{x}(t)$ es expresado en la forma de descomposición de Adomian de la siguiente manera:

$$N\mathbf{x}(t) = \sum_{k=0}^{i-1} \mathbf{A}^k(\mathbf{x}^0, \dots, \mathbf{x}^k), \quad i = 1, 2, \dots, \infty. \quad (2.38)$$

donde $\mathbf{A}^i \in \mathbb{R}^n$ son los denominados polinomios de Adomian, calculados de la siguiente manera [50]:

$$A_p^k = \frac{1}{k!} \left[\frac{d^k}{d\lambda^k} N(v_p^k(\lambda)) \right]_{\lambda=0}, \quad v_p^k(\lambda) = \sum_{u=0}^k (\lambda)^u x_p^u, \quad (2.39)$$

$$k = 0, 1, \dots, (i-1), \quad p = 1, 2, \dots, n. \quad (2.40)$$

Considerando la expresión (2.37), la solución para el sistema de orden fraccional (2.35) es la siguiente

$$\begin{aligned}
 \mathbf{x}^0(t) &= J_{t_0}^q \mathbf{g}(t) + \boldsymbol{\eta}, \\
 \mathbf{x}^1(t) &= J_{t_0}^q L \mathbf{x}^0(t) + J_{t_0}^q \mathbf{A}^0(\mathbf{x}^0(t)), \\
 &\vdots \\
 \mathbf{x}^i(t) &= J_{t_0}^q L \mathbf{x}^{i-1}(t) + J_{t_0}^q \mathbf{A}^{i-1}(\mathbf{x}^0(t), \dots, \mathbf{x}^{i-1}(t)), \\
 \mathbf{x}(t) &= \sum_{k=0}^i \mathbf{x}^k(t).
 \end{aligned} \tag{2.41}$$

Cabe señalar que la solución anterior presenta un radio de convergencia bastante pequeño en relación con la trayectoria real del sistema (2.35), mediante simulaciones numéricas es fácil verificar que difiere de la verdadera solución en largos periodos de tiempo. Para solventar este problema, el estudio de la trayectoria en un espacio de tiempo $[t_0, t]$ se realiza en M subintervalos $[t_0, t_1], [t_1, t_2], \dots, [t_{M-1}, t_M]$, divididos por un tamaño de paso $h = (t_M - t_{M-1})$ de esta forma, los valores para cada intervalo de tiempo son calculados a partir de un valor pasado.

Dada la explicación anterior y de acuerdo con las propiedades (2.10) y (2.12), la solución para el sistema de orden fraccional (2.35) es reescrita en la siguiente forma:

$$\begin{aligned}
 \mathbf{x}^0(t_j) &= \frac{h^q}{\Gamma(q+1)} \mathbf{g}(t) + \boldsymbol{\eta}, \\
 \mathbf{x}^1(t_j) &= L \mathbf{x}^0(t_j) + \mathbf{A}^0(\mathbf{x}^0(t_j)), \\
 &\vdots \\
 \mathbf{x}^i(t_j) &= L \mathbf{x}^{i-1}(t_j) + \mathbf{A}^{i-1}(\mathbf{x}^0(t_j), \dots, \mathbf{x}^{i-1}(t_j)), \\
 \mathbf{x}(t_{j+1}) &= \sum_{k=0}^i \mathbf{x}^k(t_j) \frac{h^k q}{\Gamma(kq+1)},
 \end{aligned} \tag{2.42}$$

con $j = 1, 2, \dots, M$.

2.7. Hardware embebido

El hardware embebido se refiere a la arquitectura informática- electrónica de los sistemas embebidos. Este se encuentra diseñado digitalmente para ser flexibles y cumplir

con una amplia gama de aplicaciones. Por lo general, este tipo de hardware integra un procesador asociado a una amplia variedad de dispositivos de entrada y salida, siendo el software el que dicta la orientación de la aplicación.

2.7.1. Filosofía de diseño RISC

Actualmente se puede diferenciar entre dos filosofías de diseño de arquitecturas de procesamiento: Computador de instrucciones reducido (RISC) y computador de un conjunto de instrucciones complejo (CISC) [51]. Las principales características que presenta una arquitectura RISC son:

- Un conjunto limitado y simple de instrucciones. Se cuenta con un conjunto constituido por instrucciones capaces de ejecutarse en un ciclo de reloj.
- Instrucciones orientadas a los registros con acceso limitado a memoria. ofrece pocas instrucciones básicas “LOAD” y “Store” que pueden ingresar a la memoria. El resto de ellas operan exclusivamente con registros.
- Un gran banco de registros. Los procesadores de tipo RISC contienen muchos registros de manera que las variables y los resultados intermedios usados durante la ejecución del programa no requieran utilizar la memoria.

En resumen RISC, es una filosofía que se concentra en reducir la complejidad de las instrucciones realizadas por el hardware, dado que se requiere proporcionar una mayor flexibilidad e inteligencia en el software. Este tipo de arquitectura está destinada a ofrecer instrucciones simples pero potentes que se ejecutan en un solo ciclo a una gran velocidad de reloj. Por otro lado, en los últimos años los sistemas integrados ha venido incrementando su potencial, tanto de procesamiento como de integración en un chip para dar la posibilidad a sistemas más complejos. Los sistemas integrados en Chip (SoCs), son diseños complejos que ponen todo el sistema en un solo piso de silicio en lugar de placas de circuitos impresos que contienen una gran cantidad de Chips. Cuentan con procesadores integrados, subsistemas de memoria y periféricos específicos de la aplicación requerida [52].

2.7.2. Plataformas ARM

La filosofía RISC ha tenido influencia en la arquitectura de numerosos procesadores [51], incluyendo la arquitectura ARM (Maquina tipo RISC avanzada). Aunque existen varios procesadores RISC empleados en SoC, como son: AVR, ALPHA SPARC, UltraS-PARK, OpenRISC, etc. Hoy en día los procesadores ARM tienen una gran presencia en dispositivos electrónicos de consumo, tales como: smart-phones, tablets, computadoras, smart-TVs y en otros dispositivos de desarrollo tales como Arduino, FPGAs y Rasp-Berrys.

Considerando la motivación anterior, el diseño de osciladores caóticos que se plantea en esta tesis, se desarrolla en una plataforma ARM SoC. Este enfoque consiste en una micro-computadora que se encarga de procesar los datos digitales, similar a los DSPs (procesadores de señales digitales) los procesadores ARM realizan de forma rápida y repetida operaciones aritméticas basadas en formato de punto flotante de precisión simple o doble, las instrucciones pueden ser programadas desde lenguaje de bajo nivel ensamblador y dependiendo del fabricante es posible programar en lenguajes estructurados de alto nivel como C o Python.

2.7.3. Raspberry Pi

Raspberry Pi es un tipo de computador de bajo costo basado en ARM SoC. Estas tarjetas cuentan con interfaces de hardware que permiten la integración con otros dispositivos electrónicos, una gran cantidad de librerías de propósitos específicos y una gran comunidad desarrollando aplicaciones [19, 53, 54]. Para este trabajo se utiliza una tarjeta “Raspberry Pi 3 B+” (R-Pi 3 B+) basada en un ARM Cortex A53 de 64 bits. La computadora R-Pi 3 B+ y su arquitectura se ilustran en la figura 2.2.

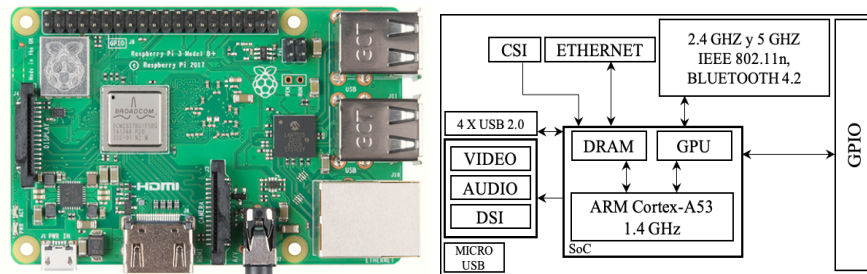


Figura 2.2: Tarjeta y arquitectura de la computadora SoC Raspberry Pi 3 modelo B+.

En la tabla 2.1 se presenta una breve descripción de las capacidades de hardware de esta tarjeta.

SoC, CPU	Bradcom BCM2837B0, Cortex-A53(ARMv8) 64 bits
GPU	VideoCore IV 400 MHz
Frecuencia de reloj	1.4 GHz
Memoria (DRAM)	1 GB LPDDR2
Conexión de red	Ethernet RJ45
Conectividad inalámbrica	2.4 GHz IEEE y 5 GHz 802.11 n, Bluetooth 4.2
PUERTOS	GPIO 40 pines
	HDMI
	4 × USB 2.0
	CSI (cámara Raspberry Pi)
	DSI (pantalla táctil)
	Micro SD
	Micro USB (Alimentación 5.1 v)
	Audio Jack 3.5 mm

Tabla 2.1: Especificaciones de la tarjeta Raspberry Pi 3 modelo B+.

Capítulo 3

Análisis y simulación de osciladores caóticos de orden entero y fraccional

3.1. Oscilador caótico de Chen

En 1999, G. Chen encontró un atractor caótico en un sistema autónomo tridimensional [55], como se observa en la referencia [56] el sistema de Chen pertenece a una clase particular de sistemas caóticos que, junto al sistema Lorenz y de Lü comparten ciertas características, tales como: número de parámetros, propiedades en sus puntos de equilibrio, simetría y disipatividad pero con una topología diferente [57], siendo el sistema de Chen quien presenta una dinámica más compleja. Es por tal motivo la importancia de su estudio en esta tesis.

3.1.1. Modelo de orden entero

El sistema de Chen de orden entero se encuentra representado de la siguiente manera

$$\begin{aligned}\dot{x} &= a(x_2 - x_1), \\ \dot{y} &= (c - a)x_1 - x_1x_3 + cx_2, \\ \dot{z} &= x_1x_2 - bx_3,\end{aligned}\tag{3.1}$$

donde a , b y c son parámetros reales positivos.

Los puntos de equilibrio del sistema (3.1) se obtienen resolviendo el siguiente sistema de ecuaciones

$$\begin{aligned} a(x_2 - x_1) &= 0, \\ (c - a)x_1 - x_1x_3 + cx_2 &= 0, \\ x_1x_2 - bx_3 &= 0. \end{aligned} \tag{3.2}$$

De la expresión anterior se encuentran los siguientes puntos de equilibrio $(\bar{x}_1, \bar{x}_2, \bar{x}_3)$:

$$\begin{aligned} Pe_1 &= (0, 0, 0), \\ Pe_2 &= (\sqrt{b(2c - a)}, \sqrt{b(2c - a)}, 2c - a), \\ Pe_3 &= (-\sqrt{b(2c - a)}, -\sqrt{b(2c - a)}, 2c - a). \end{aligned} \tag{3.3}$$

Sustituyendo (3.3) por los valores correspondientes

$$a = 35, \quad b = 3, \quad c = 28, \tag{3.4}$$

se obtienen los siguientes resultados numéricos

$$\begin{aligned} Pe_1 &= (0, 0, 0), \\ Pe_2 &= (7.9373, 7.9373, 21), \\ Pe_3 &= (-7.9373, -7.9373, 21). \end{aligned} \tag{3.5}$$

Ahora se procede a determinar si los puntos equilibrio encontrados son hiperbólicos o no hiperbólicos. Considere la matriz jacobiana del sistema (3.1)

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} -a & a & 0 \\ c - a - x_3 & c & -x_1 \\ x_2 & x_1 & -b \end{bmatrix}, \tag{3.6}$$

para la cual se obtiene la ecuación característica $|\lambda I - \mathbf{J}(\bar{x}_1, \bar{x}_2, \bar{x}_3)| = 0$, para cada punto de equilibrio Pe_1 , Pe_2 y Pe_3 (3.5).

En Pe_1

$$|\lambda I - \mathbf{J}(0, 0, 0)| = \lambda^3 + 10\lambda^2 - 714\lambda - 2205 \tag{3.7}$$

donde

$$\begin{aligned}\lambda_1 &= -30.8359, \\ \lambda_2 &= 23.8359, \\ \lambda_3 &= -3.\end{aligned}\tag{3.8}$$

En Pe_2 y Pe_3

$$\begin{aligned}|\lambda I - \mathbf{J}(7.9373, 7.9373, 21)| &= |\lambda I - \mathbf{J}(-7.9373, -7.9373, 21)| \\ &= \lambda^3 + 10\lambda^2 + 84\lambda + 4410,\end{aligned}\tag{3.9}$$

donde

$$\begin{aligned}\lambda_1 &= -18.4280, \\ \lambda_2 &= 4.2140 + 14.8847i, \\ \lambda_3 &= 4.2140 - 14.8847i.\end{aligned}$$

De los valores propios obtenidos, se observa que existe un $Re(\lambda_i) > 0$ para algún j , y $Re(\lambda_k) < 0$ para algún $k \neq j$. Por lo tanto, los puntos de equilibrio (3.5) son hiperbólicos, de acuerdo con el **teorema de Hartman-Grobman** se garantiza que la estabilidad estos puntos es preservada cuando se linealiza el sistema (3.1) alrededor de estos mismos.

A partir del análisis realizado se deduce lo siguiente:

- A P_1 le corresponden valores propios $\lambda_{1,3} < 0$ y $\lambda_2 > 0$ correspondientes a un punto de equilibrio tipo silla de índice uno.
- Los puntos de equilibrio P_2 y P_3 les corresponden valores propios que satisfacen $\lambda_1 < 0$ y $\lambda_{2,3} = \alpha \pm \beta i$ con $\alpha > 0$ y $\beta \neq 0$. Con lo que se puede determinar que estos puntos de equilibrio son de tipo silla de índice dos.

Por tanto se puede determinar que al atractor caótico le corresponden dos enrollamientos formados alrededor de los puntos de equilibrio Pe_2 y Pe_3 , los cuales están unidos por el punto de equilibrio Pe_1 .

La solución para del sistema de Chen (3.1) utilizando el método de Runge-Kutta

de cuarto orden (2.30), tiene la siguiente forma

$$\begin{aligned} x_1(t_{k+1}) &= x_1(t_k) + \frac{1}{6}(k\theta_1 + 2k\theta_2 + 2k\theta_3 + hk\theta_4), \\ x_2(t_{k+1}) &= x_2(t_k) + \frac{1}{6}(k\phi_1 + 2k\phi_2 + 2k\phi_3 + hk\phi_4), \\ x_3(t_{k+1}) &= x_3(t_k) + \frac{1}{6}h(k\rho_1 + 2k\rho_2 + 2k\rho_3 + hk\rho_4). \end{aligned} \quad (3.10)$$

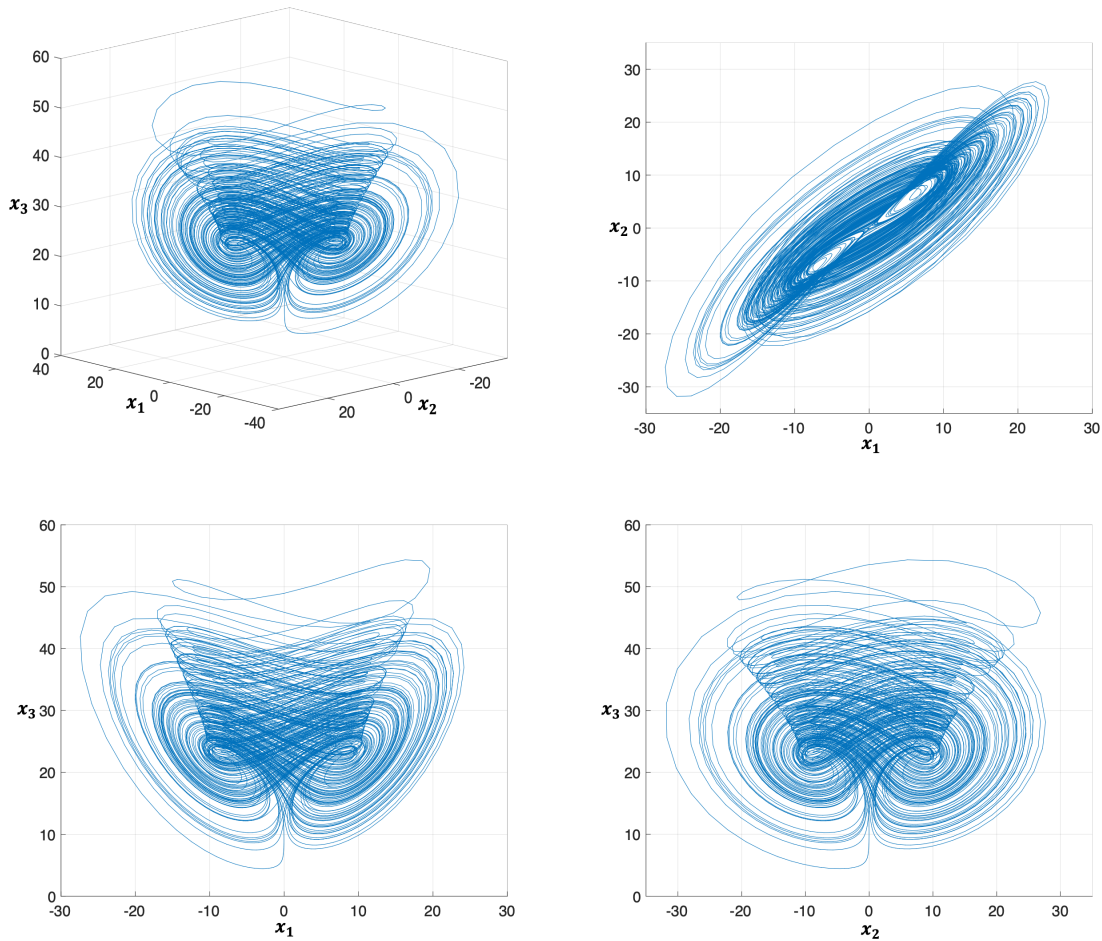


Figura 3.1: Resultado de la simulación del sistema de Chen de orden entero, realizada con tiempo de simulación $T_{sim} = 90$ s, paso de integración $h = 0.008$ y condiciones iniciales $(x_1(t_0), x_2(t_0), x_3(t_0)) = (1, 0, 30)$.

donde

$$\begin{aligned}
k\theta_1 &= a(x_2(t_k) - x_1(t_k)), \\
k\phi_1 &= (c - a)x_1(t_k) - x_1(t_k)x_3(t_k) + cx_2(t_k), \\
k\rho_1 &= x_1(t_k)x_2(t_k) - bx_3(t_k), \\
k\theta_2 &= a\left((x_2(t_k) + \frac{h}{2}k\phi_1) - (x_1(t_k) + \frac{h}{2}k\theta_1)\right), \\
k\phi_2 &= (c - a)(x_1(t_k) + \frac{h}{2}k\theta_1) - (x_1(t_k) + \frac{h}{2}k\theta_1)(x_3(t_k) + \frac{h}{2}k\rho_1) + c(x_2(t_k) + \frac{h}{2}k\phi_1), \\
k\rho_2 &= (x_1(t_k) + \frac{h}{2}k\theta_1)(x_2(t_k) + \frac{h}{2}k\phi_1) - b(x_3(t_k) + \frac{h}{2}k\rho_1), \\
k\theta_3 &= a\left((x_2(t_k) + \frac{h}{2}k\phi_2) - (x_1(t_k) + \frac{h}{2}k\theta_2)\right), \\
k\phi_3 &= (c - a)(x_1(t_k) + \frac{h}{2}k\theta_2) - (x_1(t_k) + \frac{h}{2}k\theta_2)(x_3(t_k) + \frac{h}{2}k\rho_2) + c(x_2(t_k) + \frac{h}{2}k\phi_2), \\
k\rho_3 &= (x_1(t_k) + \frac{h}{2}k\theta_2)(x_2(t_k) + \frac{h}{2}k\phi_2) - b(x_3(t_k) + \frac{h}{2}k\rho_2), \\
k\theta_4 &= a\left((x_2(t_k) + hk\phi_3) - (x_1(t_k) + hk\theta_3)\right), \\
k\phi_4 &= (c - a)(x_1(t_k) + hk\theta_3) - (x_1(t_k) + hk\theta_3)(x_3(t_k) + hk\rho_3) + c(x_2(t_k) + hk\phi_3), \\
k\rho_4 &= (x_1(t_k) + hk\theta_3)(x_2(t_k) + hk\phi_3) - b(x_3(t_k) + hk\rho_3).
\end{aligned}$$

En la figura 3.1 se presentan los resultados de simulación en MATLAB a partir de la solución (3.10).

3.1.2. Modelo de orden fraccional

El sistema de Chen de orden fraccional se encuentra representado de la siguiente manera [47]:

$$\begin{aligned}
D_t^q x_1 &= a(x_2 - x_1), \\
D_t^q x_2 &= (c - a)x_1 - x_1x_3 + cx_2, \\
D_t^q x_3 &= x_1x_2 - bx_3,
\end{aligned} \tag{3.11}$$

donde D_t^q es el operador derivativo de Grünwald-Letnikov, a , b y c son parámetros reales positivos, definidos en (3.4). Como se puede observar la variación en este modelo de Chen, es exclusivo en su orden derivativo $0 < q < 1$, por lo tanto, garantiza que los

puntos de equilibrio determinados en la expresión (3.5), son consistentes con los puntos de equilibrio en el sistema de Chen de orden fraccional.

En la tabla 3.1 se presentan los puntos de equilibrio del sistema (3.11) con sus respectivos valores propios. Para realizar el análisis de estabilidad de sus puntos de equilibrio,

Punto de equilibrio $(\bar{x}_1, \bar{x}_2, \bar{x}_3)$	Valores propios $\lambda_1, \lambda_2, \lambda_3,$
$Pe_1 = (0, 0, 0)$	$\lambda_1 = -30.8359$ $\lambda_2 = 23.8359$ $\lambda_3 = -3$
$Pe_{2,3} = (\pm 7.9373, \pm 7.9373, 21)$	$\lambda_1 = -18.4280$ $\lambda_2 = 4.2140 + 14.8847i$ $\lambda_3 = 4.2140 - 14.8847i$

Tabla 3.1: Puntos de equilibrio del sistema de Chen de orden fraccional y sus respectivos valores propios.

es conveniente determinar el mínimo orden fraccional q , donde el sistema de Chen presenta un comportamiento caótico. Considerando la expresión (2.20) y sustituyendo por los valores propios correspondientes a los puntos de equilibrio Pe_2 y Pe_3 , el mínimo orden fraccional q es el siguiente:

$$q > \frac{2}{\pi} \arctan \left(\frac{|\pm 14.8847|}{4.2140} \right), \quad (3.12)$$

$$q > 0.8244.$$

A partir de este resultado, se tiene el intervalo conocido $0.8244 < q \leq 1$, donde el sistema de Chen presenta un comportamiento caótico.

Sin pérdida de generalidad, se considera un valor arbitrario $q \in (0.8244, 1)$. Sea entonces $q = 0.9$, se procede a determinar la estabilidad de los puntos de equilibrio.

Estabilidad en $Pe_1 = (0, 0, 0)$:

$$\begin{bmatrix} |\arctan(\lambda_1)| \\ |\arctan(\lambda_2)| \\ |\arctan(\lambda_3)| \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad (3.13)$$

$$|\arctan(\lambda_i)| < q \frac{\pi}{2}, \quad i = 1, 2, 3.$$

Estabilidad en $Pe_{2,3} = (\pm 7.9373, \pm 7.9373, 21)$:

$$\begin{bmatrix} |\arctan(\lambda_1)| \\ |\arctan(\lambda_2)| \\ |\arctan(\lambda_3)| \end{bmatrix} = \begin{bmatrix} 0 \\ 1.29 \\ 1.29 \end{bmatrix}, \quad (3.14)$$

$$|\arg(\lambda_i)| < q \frac{\pi}{2}, \quad i = 1, 2, 3.$$

Con esto se demuestra que los puntos de equilibrio son inestables.

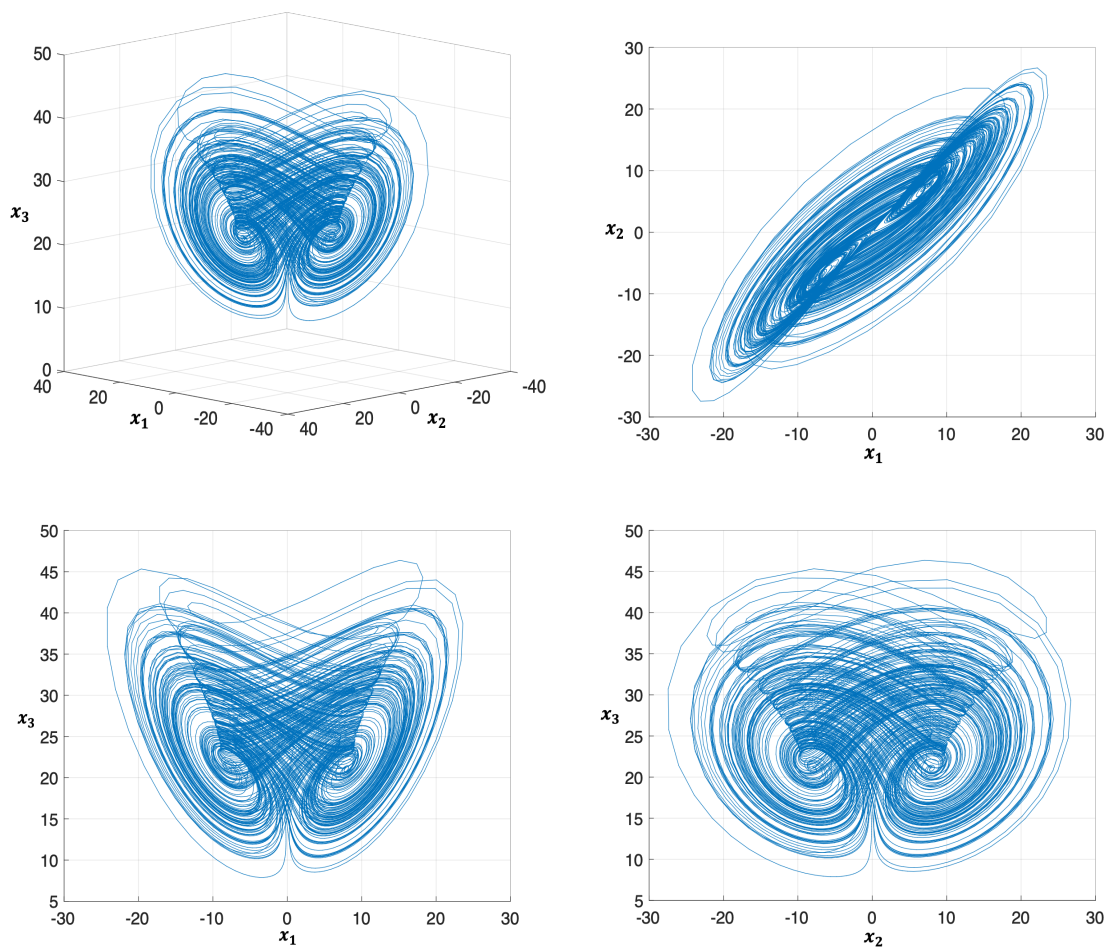


Figura 3.2: Resultado de la simulación del sistema de Chen de orden fraccional $q = 0.9$, realizada con tiempo de simulación $T_{sim} = 90$ s, paso de integración $h = 0.007$ y condiciones iniciales $(x_1(t_0), x_2(t_0), x_3(t_0)) = (1, 0, 25)$.

La solución numérica del sistema de Chen de orden fraccionario por el método de

Grünwald-Letnikov (2.32), tiene la siguiente forma:

$$\begin{aligned}
C_0^q &= 1, \quad C_j^q = \left(1 - \frac{1+q}{j}\right) C_{j-1}^q, \quad j = 1, 2, \dots, k, \\
x_1(t_{k+1}) &= a(x_2(t_k) - x_1(t_k))h^q - \sum_{j=1}^k C_j^q x_1(t_{k-j}), \\
x_2(t_{k+1}) &= \left((c-a)(x_1(t_k) - x_1(t_k)x_3(t_k) + cx_2(t_k))\right)h^q - \sum_{j=1}^k C_j^q x_2(t_{k-j}), \\
x_3(t_{k+1}) &= (x_1(t_k)x_2(t_k) - bx_3(t_k))h^q - \sum_{j=1}^k C_j^q x_3(t_{k-j}).
\end{aligned} \tag{3.15}$$

La solución numérica (3.15) corresponde a los retratos de fase presentados en la figura 3.2. De forma cualitativa es fácil observar que, bajo un mismo conjunto de parámetros, el espacio de estados del sistema de Chen de orden fraccional $q = 0.9$ es geográficamente constreñido respecto al sistema de Chen de orden entero.

3.2. Oscilador caótico de Múltiples-Enrollamientos

Dentro de los métodos existentes para generar señales caóticas, se encuentra el acoplamiento de funciones lineales por segmentos a sistemas dinámicos. Una función lineal por segmentos (PWL) es una función que envuelve a la variable de estado x y se denota como $f(x; \cdot)$. Se encuentra formada por un conjunto de n regiones lineales, cuya membresía se encuentra definida por alguna condición, que no necesariamente incluye la variable de estado. El objetivo de estas funciones es modelar señales cuyo comportamiento complejo no permite ser expresado de forma analítica. Un ejemplo clásico de función PWL es la función valor absoluto $|x|$, definida de la siguiente manera

$$f(x) = \begin{cases} x(t), & x(t) \leq 0, \\ -x(t), & x(t) > 0. \end{cases} \tag{3.16}$$

Se encuentra reportado que la aproximación de no linealidades de un sistema vía funciones PWL, preserva fenómenos importantes de la región que ha sido modelada. De esta forma la dinámica puede ser reconstruida y estudiada sobre un cierto rango de parámetros, como ejemplo se tiene la referencia [58] donde se propuso un nuevo modelo de célula beta basada en aproximaciones PWL.

En particular las funciones PWL han sido ampliamente utilizadas para generar caos: en la referencia [59] se propuso un sistema caótico de Chen PWL; está reportado que el acoplamiento de funciones PWL en sistemas que en principio presentan una dinámica estable generan caos, tal es el caso del sistema caótico de Múltiples Enrollamientos que se estudia en esta tesis, la complejidad de la dinámica puede incrementarse para generar atractores en orientaciones 1D, 2D, 3D, mediante conmutación de control [60], en sistemas ampliamente estudiados como el sistema caótico de Chua, se encuentran atractores caóticos sin puntos de equilibrio [61]. Otros sistemas caóticos desarrollados a partir de la misma técnica se presentan en [62–65].

La forma de solucionar sistemas con funciones PWL es un área de investigación abierta, incluso en sistemas que presentan una dinámica no caótica, el análisis incrementa en complejidad dadas las discontinuidades existentes entre las regiones de la función PWL.

3.2.1. Modelo de orden entero

Considere el siguiente sistema autónomo lineal de la forma $\dot{\mathbf{x}} = A\mathbf{x}$ de tercer orden.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a & -b & -c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad (3.17)$$

donde $a = 0.7$, $b = 0.7$, $c = 0.7$. Al resolver el siguiente sistema de ecuaciones

$$\begin{aligned} x_2 &= 0, \\ x_3 &= 0, \\ -ax_1 - bx_2 - cx_3 &= 0, \end{aligned} \quad (3.18)$$

se hace evidente que el sistema (3.17) tiene un solo punto de equilibrio en el origen $Pe_1(0, 0, 0)$. Cuya ecuación característica $|\lambda I - A\mathbf{x}| = 0$, tiene la siguiente forma:

$$\lambda^3 + c\lambda^2 + b\lambda + a = 0, \quad (3.19)$$

sustituyendo en (3.19) los valores de parámetros correspondientes, se obtienen los siguientes valores propios

$$\begin{aligned}\lambda_1 &= -8480, \\ \lambda_2 &= 0.0740 + 0.9055i, \\ \lambda_3 &= 0.0740 - 0.9055i.\end{aligned}\tag{3.20}$$

Los valores propios obtenidos satisfacen un punto silla de índice dos, por lo tanto, las trayectorias de (3.17) divergen de forma oscilatoria cuando $t \rightarrow \infty$. Se hace evidente que la región de atracción caótica es inexistente. Ahora, considere la siguiente función en series saturadas

$$f(x_1; k_s, h_s, p_s, q_s) = \begin{cases} (2q_s + 1)k_s & x_1 > q_s h_s + 1, \\ k_s(x_1 - i h_s) + 2i k_s & |x_1 - i h_s| \leq 1, \\ & -p_s \leq i \leq q_s, \\ (2i + 1)k_s & i h_s + 1 < x_1 < (i + 1)h_s - 1, \\ & -p_s \leq i \leq q_s - 1, \\ -(2p_s + 1)k_s & x_1 < -p_s h_s - 1, \end{cases}\tag{3.21}$$

donde $k_s = 10$ es el factor multiplicativo de la pendiente, $h_s = 2k_s$ es el factor multiplicativo de las regiones saturadas; considerando $p_s = q_s = 1$ se tiene que $i_1 = -1$, $i_2 = 0$, $i_3 = 1$. Como se ilustra en la figura 3.3, la función en series saturadas (3.21) tiene 7 segmentos, para generar 4 enrollamientos en orientación 1D.

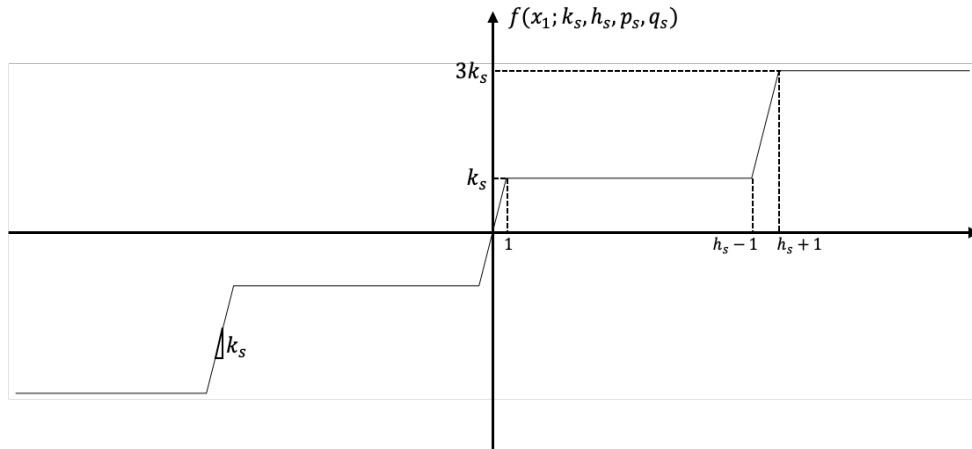


Figura 3.3: Descripción de una serie de funciones saturadas para generar 4 enrollamientos.

Sumando esta no linealidad al sistema (3.17), se obtiene la siguiente expresión

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ -a & -b & -c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix} f(x_1; k, h, p, q), \quad (3.22)$$

con $a = b = c = d = 0.7$.

En (3.22) existen $(2(p_s + q_s) + 3)$ puntos de equilibrio, y se obtienen al resolver el siguiente sistema de ecuaciones:

$$\begin{aligned} x_2 &= 0, \\ x_3 &= 0, \\ -ax_1 - bx_2 - cx_3 + df(x_1; k_s, h_s, p_s, q_s) &= 0. \end{aligned} \quad (3.23)$$

Se hace evidente que la línea de puntos de equilibrio en el eje x_1 depende de la siguiente ecuación:

$$-x_1 + f(x_1; k_s, h_s, p_s, q_s) = 0, \quad \forall a = d \in \mathbb{R} \quad (3.24)$$

y de acuerdo con (3.21), se obtienen los siguientes puntos de equilibrio:

$$\begin{aligned} Pe_1 &= (0, 0, 0), \\ Pe_{2,3} &= \pm(10, 0, 0), \\ Pe_{4,5} &= \pm(20, 0, 0), \\ Pe_{6,7} &= \pm(30, 0, 0). \end{aligned} \quad (3.25)$$

Ahora se procede a determinar si los puntos equilibrio encontrados son hiperbólicos o no hiperbólicos.

Cuando $f(x_1; k_s, h_s, p_s, q_s) = k(x - ih_s) + 2ik_s$, con $i = -1, 0, 1$ le corresponden los puntos de equilibrio: $Pe_1 = (0, 0, 0)$ y $Pe_{4,5} = \pm(20, 0, 0)$. La matriz jacobiana del sistema (3.22) toma la siguiente forma:

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 6.3 & -0.7 & -0.7 \end{bmatrix}, \quad (3.26)$$

para la cual se obtiene la ecuación característica $|\lambda I - \mathbf{J}(\mathbf{x})| = 0$, esto es

$$\lambda^3 + 0.7\lambda^2 + 0.7\lambda - 6.3 = 0 \quad (3.27)$$

donde

$$\begin{aligned} \lambda_1 &= 1.5309, \\ \lambda_2 &= -1.1154 + 1.6944i, \\ \lambda_3 &= -1.1154 - 1.6944i. \end{aligned} \quad (3.28)$$

Cuando $f(x_1; k_s, h_s, p_s, q_s) \neq k(x_1 - ih_s) + 2ik_s$, con $i = 1, 2, 3$ le corresponden los siguientes puntos de equilibrio: $Pe_{2,3} = \pm(10, 0, 0)$ y $Pe_{6,7} = \pm(30, 0, 0)$, la matriz jacobiana del sistema (3.22) toma la siguiente forma:

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0.7 & -0.7 & -0.7 \end{bmatrix}, \quad (3.29)$$

para la cual se obtiene la ecuación característica $|\lambda I - \mathbf{J}(\mathbf{x})| = 0$, esto es

$$\lambda^3 + 0.7\lambda^2 + 0.7\lambda + 0.7 = 0 \quad (3.30)$$

donde

$$\begin{aligned} \lambda_1 &= -0.8480, \\ \lambda_2 &= 0.0740 + 0.9055i, \\ \lambda_3 &= 0.0740 - 0.9055i. \end{aligned} \quad (3.31)$$

Los puntos de equilibrio que definen a la ecuación característica (3.37), les corresponden los valores propios que satisfacen $\lambda_1 > 0$ y $\lambda_{2,3} = \alpha \pm \beta i$ con $\alpha < 0$ y $\beta \neq 0$, mientras que a los puntos de equilibrio que definen a la ecuación característica (3.40), les corresponden valores propios que satisfacen $\lambda_1 < 0$ y $\lambda_{2,3} = \alpha \pm \beta i$ con $\alpha > 0$ y $\beta \neq 0$. Por lo tanto todos los puntos de equilibrio encontrados son hiperbólicos.

A partir del análisis realizado se deduce que los puntos de equilibrio P_1 y $P_{4,5}$ son inestables de tipo silla y de índice uno, mientras que los puntos de equilibrio $P_{2,3}$ y $P_{6,7}$ son inestables de tipo silla de índice dos. Además dado que el enrollamiento solo es

generado alrededor de los puntos silla de índice dos, el sistema (3.22) tiene la posibilidad de generar $(p_s + q_s + 2)$ enrollamientos para los valores adecuados de los parámetros a, b, c, d, k_s y h_s . Es necesario enfatizar que los $(p + q + 1)$ puntos de equilibrio que definen a (3.37) son responsables de conectar los $(p_s + q_s + 2)$ enrollamientos de todo el atractor caótico y cada punto corresponde a una única pendiente de la serie de funciones saturadas (3.21). Por otra parte, cada punto de equilibrio de $(p_s + q_s + 2)$ que define a (3.40), corresponde a una región saturada de (3.21) y le corresponde un único enrollamiento de todo el atractor [40].

La solución numérica del sistema de Múltiples-Enrollamientos de orden entero, por el método de RK-4 (2.30) tiene la siguiente forma:

$$\begin{aligned} x_1(t_{k+1}) &= x_1(t_k) + \frac{1}{6}(k\theta_1 + 2k\theta_2 + 2k\theta_3 + hk\theta_4), \\ x_2(t_{k+1}) &= x_2(t_k) + \frac{1}{6}(k\phi_1 + 2k\phi_2 + 2k\phi_3 + hk\phi_4), \\ x_3(t_{k+1}) &= x_3(t_k) + \frac{1}{6}(k\rho_1 + 2k\rho_2 + 2k\rho_3 + hk\rho_4), \end{aligned} \quad (3.32)$$

donde

$$\begin{aligned} k\theta_1 &= x_2(t_k), \\ k\phi_1 &= x_3(t_k), \\ k\rho_1 &= -ax_1(t_k) - bx_2(t_k) - cx_3(t_k) + df(x_1(t_k); k_s, h_s, p_s, q_s), \\ k\theta_2 &= x_2(t_k) + \frac{h}{2}k\phi_1, \\ k\phi_2 &= x_3(t_k) + \frac{h}{2}k\rho_1, \\ k\rho_2 &= -a\left(x_1(t_k) + \frac{h}{2}k\theta_1\right) - b\left(x_2(t_k) + \frac{h}{2}k\phi_1\right) - c\left(x_3(t_k) + \frac{h}{2}k\rho_1\right) + df\left(x_1(t_k) + \frac{h}{2}k\theta_1; k_s, h_s, p_s, q_s\right), \\ k\theta_3 &= x_2(t_k) + \frac{h}{2}k\phi_2, \\ k\phi_3 &= x_3(t_k) + \frac{h}{2}k\rho_2, \\ k\rho_3 &= -a\left(x_1(t_k) + \frac{h}{2}k\theta_2\right) - b\left(x_2(t_k) + \frac{h}{2}k\phi_2\right) - c\left(x_3(t_k) + \frac{h}{2}k\rho_2\right) + df\left(x_1(t_k) + \frac{h}{2}k\theta_2; k_s, h_s, p_s, q_s\right), \\ k\theta_4 &= x_2(t_k) + hk\phi_2, \\ k\phi_4 &= x_3(t_k) + hk\rho_2, \\ k\rho_4 &= -a\left(x_1(t_k) + hk\theta_3\right) - b\left(x_2(t_k) + hk\phi_3\right) - c\left(x_3(t_k) + hk\rho_3\right) + df\left(x_1(t_k) + hk\theta_3; k_s, h_s, p_s, q_s\right). \end{aligned}$$

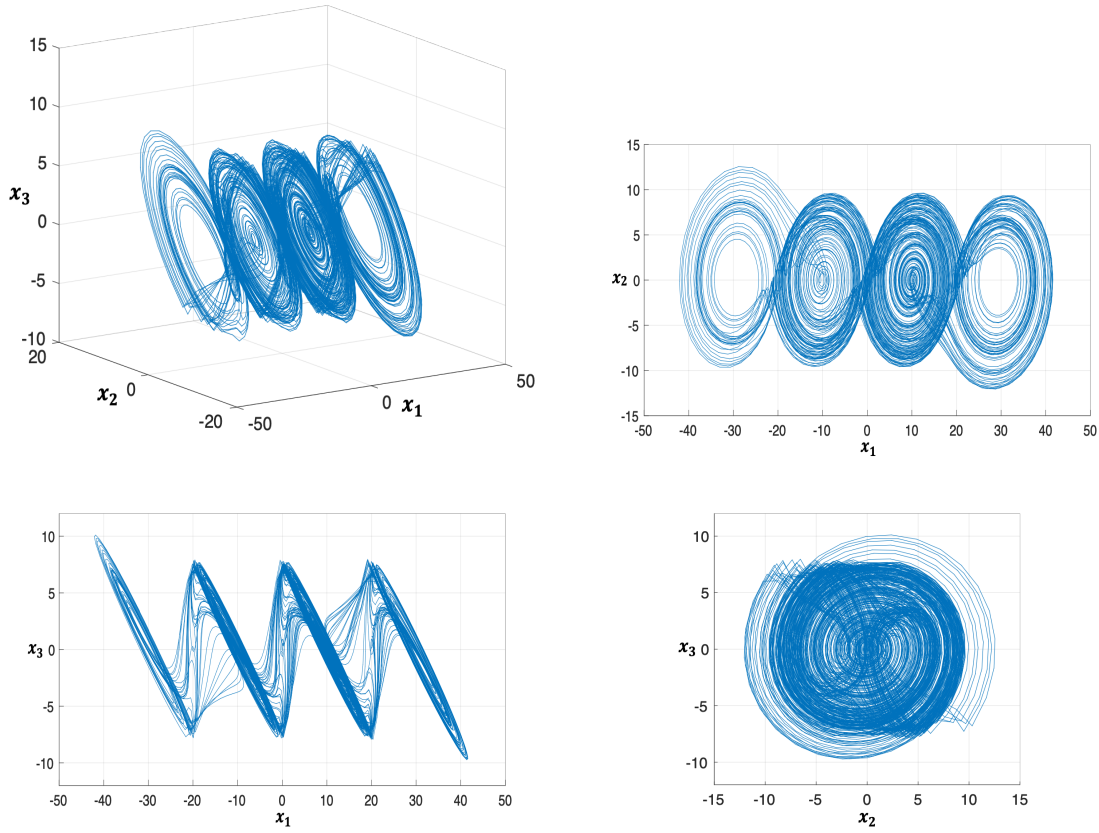


Figura 3.4: Resultado de la simulación del sistema de Múltiples Enrollamientos de orden entero, realizada con tiempo de simulación $T_{sim} = 2500$ s, paso de integración $h = 0.25$ y condiciones iniciales $(x_1(t_0), x_2(t_0), x_3(t_0)) = (1, 0, 1)$.

Puntos de equilibrio	Puntos silla de índice 1 (Pendiente)	Puntos silla de índice 2 (Saturación)
7	3	4
(0,0,0)	(0,0,0)	$\pm(10,0,0)$
$\pm(10,0,0)$	$\pm(20,0,0)$	$\pm(30,0,0)$
$\pm(20,0,0)$		
$\pm(30,0,0)$		

Tabla 3.2: Ubicación de los puntos de equilibrio del sistema (3.22) dentro de la función en series saturadas (3.21).

La solución (3.32) corresponde a los retratos de fase presentados en la figura 3.4. De forma cualitativa se observa fácilmente que el comportamiento de los puntos de equilibrio es consistente con las formulaciones teóricas presentadas en la tabla 3.2.

3.2.2. Modelo de orden fraccional

El sistema de Múltiples Enrollamientos de orden fraccional se encuentra representado de la siguiente manera.

$$\begin{aligned} D_t^q x_1 &= x_2, \\ D_t^q x_2 &= x_3, \\ D_t^q x_3 &= -ax_1 - bx_2 - cx_3 + df(x_1; k_s, h_s, p_s, q_s), \end{aligned} \quad (3.33)$$

donde D_t^q es el operador derivativo de Grünwald-Letnikov, a , b , c y d son parámetros reales positivos y $f(x_1; k, h, p, q)$ es una función en series saturadas, descrita por (3.21). De forma análoga al modelo de orden entero (3.22), el sistema de Múltiples Enrollamientos de orden fraccional tiene $(2(p_s + q_s) + 3)$ puntos de equilibrio, considerando que $a = d$ y $b = c$, estos se definen de la siguiente manera:

$$\begin{aligned} \begin{bmatrix} \bar{x}_{1_j} \\ \bar{x}_{2_j} \\ \bar{x}_{3_j} \end{bmatrix} &= \begin{bmatrix} -x_1 + f(x_1; k_s, h_s, p_s, q_s) = 0 \\ 0 \\ 0 \end{bmatrix}, \\ j &= 1, 2, \dots, (2(p_s + q_s) + 3), \forall a = d \in \mathbb{R}. \end{aligned} \quad (3.34)$$

la membresía de estos puntos de equilibrio respecto a la función en series saturadas $f(x_1; k_s, h_s, p_s, q_s)$, se encuentran descritos en la tabla 3.3. Ahora se procede a determinar la estabilidad de los puntos de equilibrio encontrados.

Considere la siguiente la variación en los parámetros

$$a = 2.2, \quad b = 1.4, \quad c = 0.6, \quad d = 2.2. \quad (3.35)$$

En $Pe_1 = (0, 0, 0)$ y $Pe_{4,5} = \pm(20, 0, 0)$. La matriz jacobiana del sistema (3.33) toma la forma

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 19.8 & -1.4 & -0.6 \end{bmatrix}, \quad (3.36)$$

para la cual se obtiene la ecuación característica

$$\lambda^3 + 0.6\lambda^2 + 1.4\lambda - 19.8 = 0 \quad (3.37)$$

donde

$$\begin{aligned} \lambda_1 &= 2.3604, \\ \lambda_2 &= -1.4802 + 2.4893i, \\ \lambda_3 &= -1.4802 - 2.4893i. \end{aligned} \quad (3.38)$$

En $Pe_{2,3} = \pm(10, 0, 0)$ y $Pe_{6,7} = \pm(30, 0, 0)$, la matriz jacobiana del sistema (3.22) toma la forma

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2.2 & -1.4 & -0.6 \end{bmatrix}, \quad (3.39)$$

para la cual se obtiene la ecuación característica

$$\lambda^3 + 0.6\lambda^2 + 1.4\lambda + 2.2 = 0 \quad (3.40)$$

donde

$$\begin{aligned} \lambda_1 &= -1.1146, \\ \lambda_2 &= 0.2573 + 1.3811i, \\ \lambda_3 &= 0.2573 - 1.3811i. \end{aligned} \quad (3.41)$$

Función en series saturadas	Puntos de equilibrio
$f(x_1; k_s, h_s, p_s, q_s) = k_s(x_1 - ih_s) + 2ik_s$, con $i = -1, 0, 1$	$Pe_1 = (0, 0, 0)$ $Pe_{4,5} = \pm(20, 0, 0)$
$f(x_1; k_s, h_s, p_s, q_s) \neq k_s(x_1 - ih_s) + 2ik_s$, con $i = -1, 0, 1$	$Pe_{2,3} = \pm(10, 0, 0)$ $Pe_{6,7} = \pm(30, 0, 0)$

Tabla 3.3: Membresía de los puntos de equilibrio del sistema (3.33) respecto a la función en series saturadas (3.21).

Ahora se procede a determinar el mínimo orden fraccional q , donde el sistema de Múltiples-Enrollamientos presenta un comportamiento caótico. Considerando la expresión (2.20), y sustituyendo por los valores propios correspondientes a los puntos de equilibrio $Pe_{1,4,5}$, el mínimo orden fraccional q se determina de la siguiente manera:

$$q > \frac{2}{\pi} \arctan \left(\frac{|\pm 1.3811|}{0.2573} \right), \quad (3.42)$$

$$q > 0.8827.$$

A partir de este resultado, se tiene el intervalo conocido $0.8827 < q \leq 1$, donde el sistema de Chen presenta un comportamiento caótico.

Sin pérdida de generalidad, se considera un valor arbitrario $q \in (0.8244, 1)$. Sea entonces $q = 0.9$, se procede a determinar la estabilidad de los puntos de equilibrio. Estabilidad en $Pe_1 = (0, 0, 0)$ y $Pe_{4,5} = \pm(20, 0, 0)$:

$$\begin{bmatrix} |\arctan(\lambda_1)| \\ |\arctan(\lambda_2)| \\ |\arctan(\lambda_3)| \end{bmatrix} = \begin{bmatrix} 0 \\ 1.0343 \\ 1.0343 \end{bmatrix} \quad (3.43)$$

$$|\arctan(\lambda_i)| < q \frac{\pi}{2}, \quad i = 1, 2, 3.$$

Estabilidad en $Pe_{2,3} = \pm(10, 0, 0)$ y $Pe_{6,7} = \pm(30, 0, 0)$:

$$\begin{bmatrix} |\arctan(\lambda_1)| \\ |\arctan(\lambda_2)| \\ |\arctan(\lambda_3)| \end{bmatrix} = \begin{bmatrix} 0 \\ 1.3866 \\ 1.3866 \end{bmatrix} \quad (3.44)$$

$$|\arctan(\lambda_i)| < q \frac{\pi}{2}, \quad i = 1, 2, 3.$$

Con esto se demuestra que los puntos de equilibrio son inestables.

La solución numérica del sistema de Múltiples Enrollamientos de orden fraccionario por el método de Grünwald-Letnikov (2.32), tiene la siguiente forma:

$$C_0^q = 1, \quad C_j^q = \left(1 - \frac{1+q}{j}\right) C_{j-1}^q, \quad j = 1, 2, \dots, k.$$

$$\begin{aligned}
x_1(t_{k+1}) &= x_2(t_k)h^q - \sum_{j=1}^k C_j^q x_1(t_{k-j}), \\
x_2(t_{k+1}) &= x_3(t_k)h^q - \sum_{j=1}^k C_j^q x_2(t_{k-j}), \\
x_3(t_{k+1}) &= \left(-ax_1(t_k) - bx_2(t_k) - cx_3(t_k) + df(x_1(t_k); k_s, h_s, p_s, q_s) \right) h^q - \sum_{j=1}^k C_j^q x_3(t_{k-j}).
\end{aligned} \tag{3.45}$$

La solución numérica (3.45) corresponde a los retratos de fase presentados en la figura 3.5. La estructura geométrica obtenida presenta una ligera deformación respecto al atractor de orden entero. No obstante, se observa que la línea de puntos de equilibrio en el eje x_1 es consistente en ambos modelos.

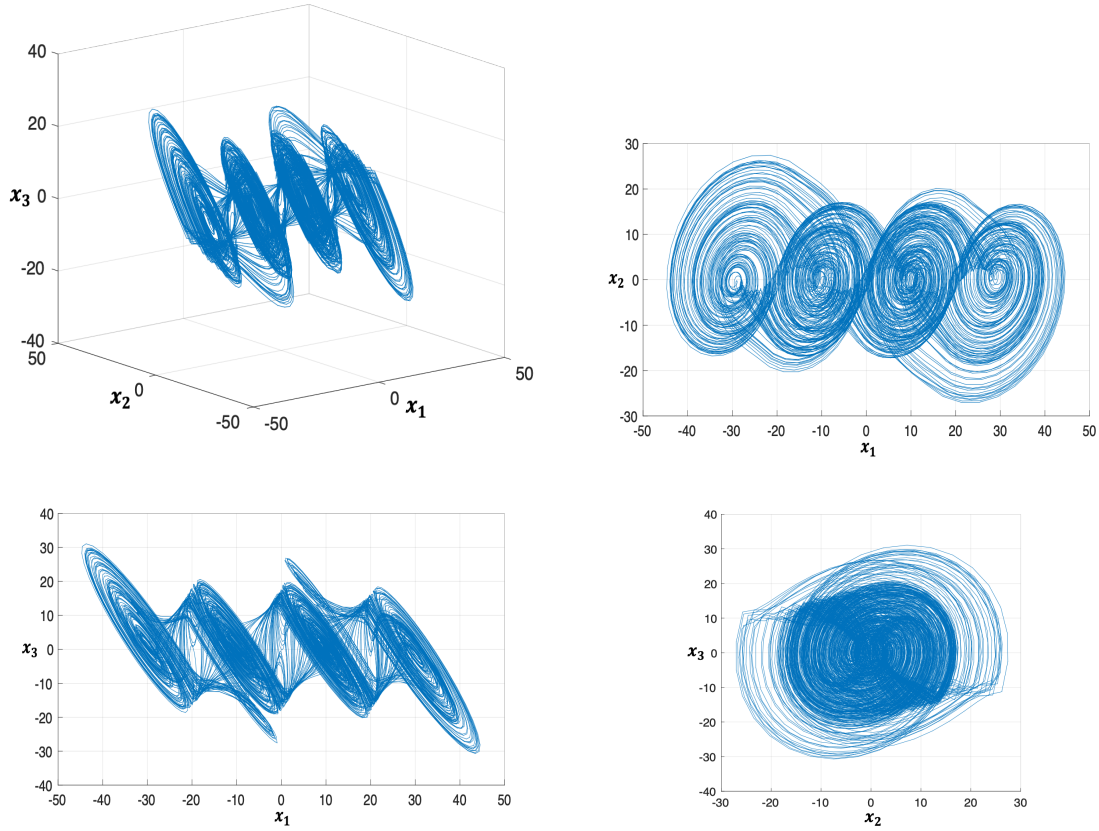


Figura 3.5: Resultado de la simulación del sistema de Múltiples Enrollamientos de orden fraccional $q = 0.9$, realizada con tiempo de simulación $T_{sim} = 3000$ s, paso de integración $h = 0.15$ y condiciones iniciales $(x_1(t_0), x_2(t_0), x_3(t_0)) = (1, 0, 1)$.

Capítulo 4

Metodología para el diseño de osciladores caóticos de orden entero y fraccional

4.1. Método 1: Runge-Kutta para sistemas de orden entero

De forma general, la implementación de sistemas caóticos de orden entero es bastante similar al análisis realizado mediante simulaciones numéricas. La principal diferencia es que el método numérico necesita ser adaptado a las especificaciones del hardware donde se plantea la realización experimental. Considerese el método de Runge-Kutta de cuarto orden

$$\begin{aligned}k_1 &= hf(t_k, x_{t_k}), \\k_2 &= hf\left(t_k + \frac{1}{2}h, x_{t_k} + \frac{1}{2}k_1\right), \\k_3 &= hf\left(t_k + \frac{1}{2}h, x_{t_k} + \frac{1}{2}k_2\right), \\k_4 &= hf(t_k + h, x_{t_k} + k_3), \\x(t_{k+1}) &= x(t_k) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4).\end{aligned}\tag{4.1}$$

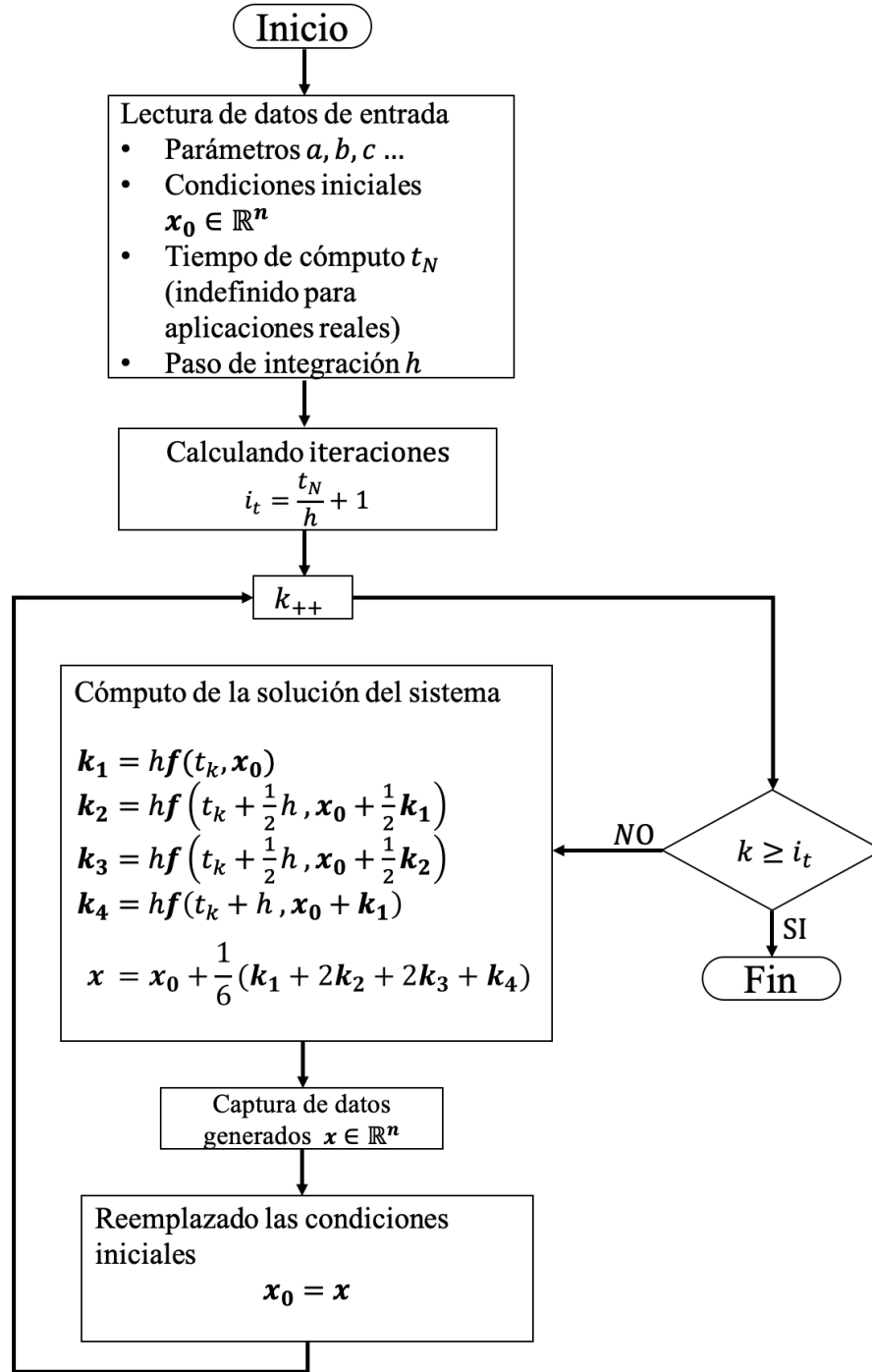


Figura 4.1: Diagrama de flujo para resolver sistemas caóticos en hardware embebido por el método Runge-Kutta .

Al implementar la expresión (4.1) en algún software matemático como MATLAB, de forma explícita el método almacena los puntos que forman la trayectoria de la solución

para intervalo de tiempo $[0, t_N]$, en el vector $x(t_k) \in \mathbb{R}^{k+1}$. No obstante, este método obtiene la solución $(x(t_1))$ para un problema de condición inicial $(x(t_0)) \in \mathbb{R}$, de forma consecutiva, para determinar la solución en $x(t_2)$ la condición inicial es reemplazada por $x(t_1)$. En este sentido, lo más conveniente es utilizar la menor memoria física disponible en el hardware embebido al momento de resolver el modelo matemático.

Considerando la explicación anterior, el esquema metodológico para implementar sistemas caóticos de orden entero se ilustra en el diagrama de flujo de la figura 4.1. Cabe señalar que este esquema no compromete la manera de resolver el sistema respecto al método (4.1), por lo tanto no existe una alteración en el resultado numérico de los sistemas de orden entero (3.1) y (3.22).

4.2. Método 2: Grünwald-Letnikov para sistemas de orden fraccional

Considere el método de Grünwald-Letnikov con principio de memoria corta

$$x(t_{k+1}) = f(t_k, x(t_k))h^q - \sum_{j=1}^v C_j^q x(t_{k-j}), \quad (4.2)$$

donde v y C_0^q están definidos de la siguiente manera

$$v = \begin{cases} 1 & k < \frac{Lm}{h} \\ \frac{Lm}{h} & k \geq \frac{Lm}{h} \end{cases}$$

$$C_0^q = 1, \quad C_j^q = \left(1 - \frac{1+q}{j}\right) C_{j-1}^q, \quad j = 1, 2, \dots, \frac{Lm}{h}. \quad (4.3)$$

Al analizar la expresión (4.2), se observa que la función de suma puede ser representado como la siguiente operación de producto punto:

$$\sum_{j=1}^v C_j^q x(t_{k-j}) = \underbrace{\begin{bmatrix} C_1^q & C_2^q & C_1^q & C_3^q & \dots & C_v^q \end{bmatrix}}_{C^q \in \mathbb{R}^{1 \times v}} \underbrace{\begin{bmatrix} x(t_{k-1}) \\ x(t_{k-2}) \\ x(t_{k-3}) \\ \vdots \\ x(t_{k-v}) \end{bmatrix}}_{x \in \mathbb{R}^v}, \quad (4.4)$$

de forma aparente el vector $x \in \mathbb{R}^v$ tiene una dimensión fija, cabe señalar sus elementos mantienen una fuerte dependencia al tiempo discreto, esto implica que el valor de los elementos $x(t_{k-1}), x(t_{k-2}), \dots, x(t_{k-v})$, necesitan ser actualizados para cada intervalo subintervalo $[t_k, t_{k+1}]$. por otra parte el valor definido o indefinido de t_k está sujeto al diseñador. Desde el punto de vista computacional, el método de Grünwald-Letnikov presentado en (4.2) no elimina las soluciones anteriores a $x(t_{k-v})$ que ya han sido utilizadas. En consecuencia, obtenemos un vector de soluciones $x(t_k) \in \mathbb{R}^{k+1}$, con una dimensión tan alta como el número de iteraciones tiende al infinito. La figura 4.2 ilustra este problema.

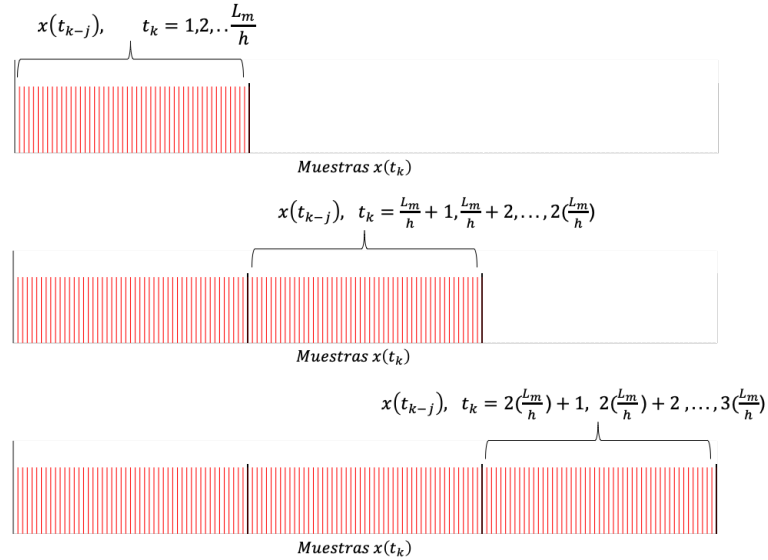


Figura 4.2: Enfoque numérico usual para resolver sistemas de orden fraccional con el método de Grünwald-Letnikov. Los paréntesis denotan solo las muestras que son utilizadas en el principio de memoria corta, no obstante, las muestras anteriores son almacenadas para todo el tiempo de simulación.

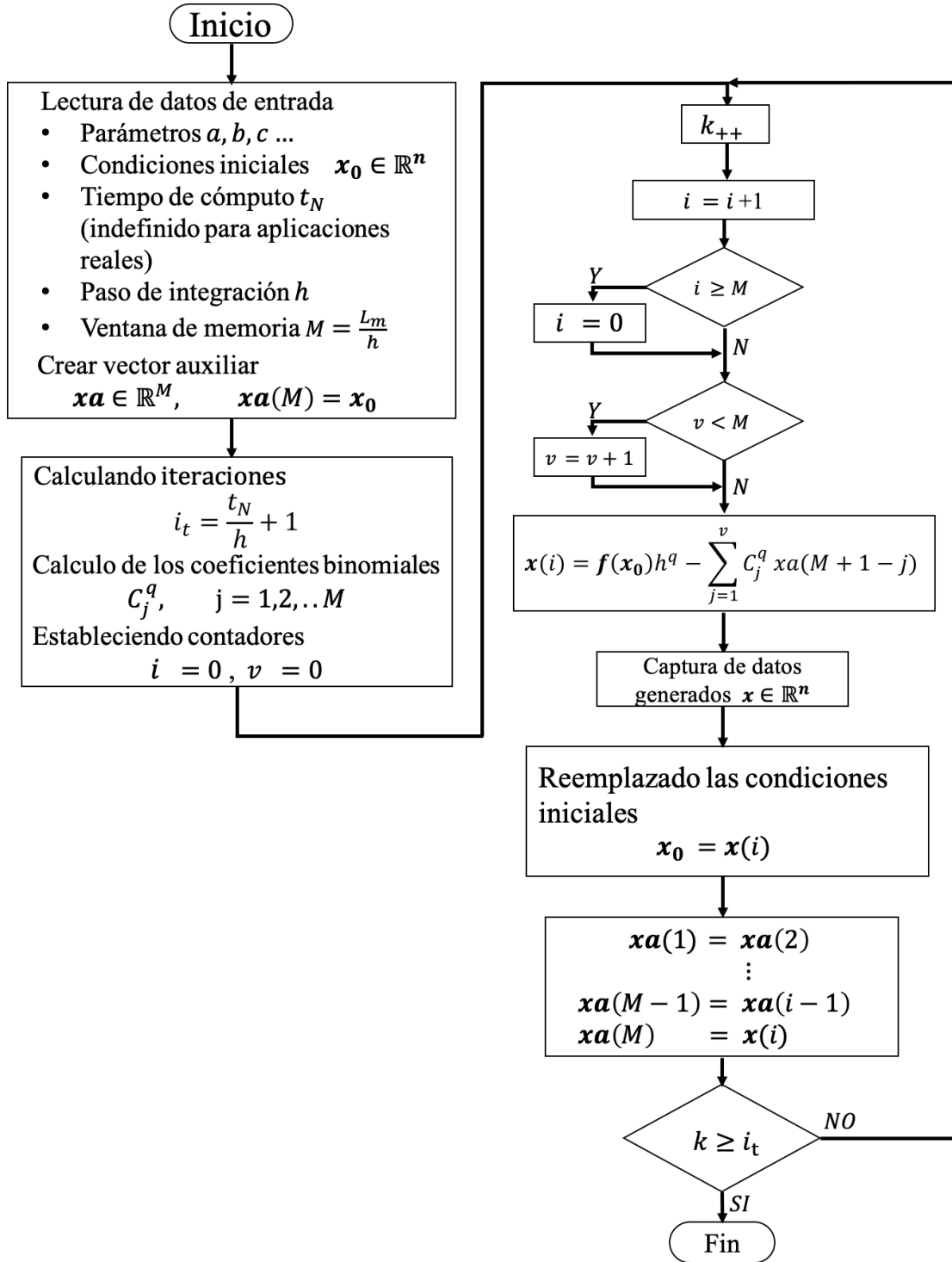


Figura 4.3: Diagrama de flujo para resolver sistemas caóticos en hardware embebido por el método Grünwald-Letnikov.

Un aspecto crucial en las plataformas digitales basadas en procesadores son los re-

cursores computacionales utilizados por un programa. Cabe mencionar que en general, estas plataformas tienen limitados recursos de hardware, tal como la memoria física dedicada a almacenar datos temporales. Además, el procesamiento de una gran cantidad puede reducir el rendimiento del cálculo computacional a medida que pasa el tiempo. Por lo tanto, las expresiones (4.2) y (4.2) no son convenientes para una implementación en hardware embebido. Considerando los motivos anteriores, se propone el esquema metodológico presentado en la figura 4.3. De forma específica, este esquema elimina las muestras anteriores a $x(t_k - v)$. En consecuencia el almacenamiento de datos para el vector $x \in \mathbb{R}^v$, mantiene una dimensión fija determinada por $\frac{L_m}{h}$, cabe señalar que la eliminación de estas muestras no compromete la generación del atractor caótico. El nuevo enfoque numérico se ilustra en la figura 4.4.

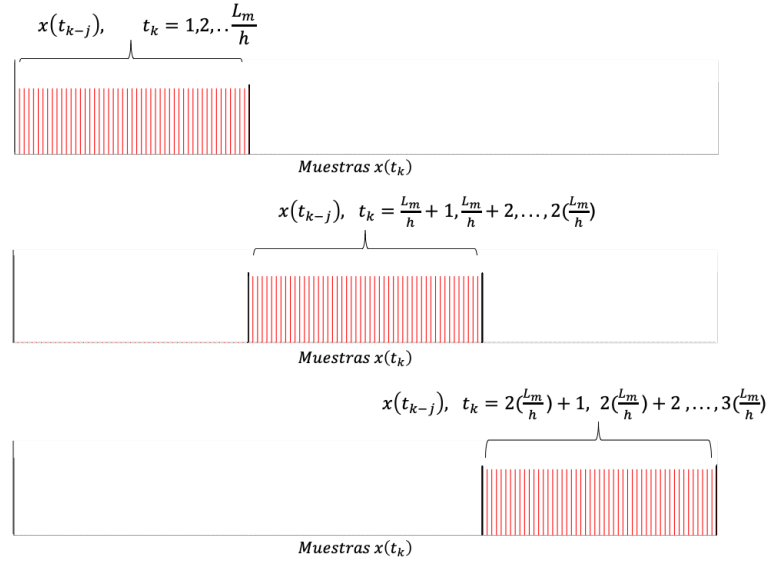


Figura 4.4: El enfoque numérico propuesto para implementar el principio de memoria corta en el método de Grünwald-Letnikov. Los paréntesis denotan las muestras utilizadas para determinar la solución en $x(t_{k+1})$.

De forma general, la aplicación del principio de memoria corta en el método de Grünwald Letnikov tiene como consecuencia una reducción en la aproximación de la solución del sistema caótico. Para reducir tal efecto se requiere de una longitud de memoria L_m apropiada, cabe señalar que en la practica no existe una formulación teórica para determinar este valor, por este motivo se realiza un análisis de barrido en función la longitud de memoria, dicho análisis consiste en resolver el sistema utilizando diferentes valores para L_m , hasta obtener un resultado consistente. En la siguiente

subsección se presenta dicho barrimiento para los sistemas de Chen y de Múltiples Enrollamientos de orden fraccional.

4.2.1. Oscilador caótico de Chen de orden fraccional basado en Grünwald-Letnikov

La solución numérica del sistema de Chen de orden fraccional, considerando el principio de memoria corta en el método de Grünwald-Letnikov tiene la siguiente forma

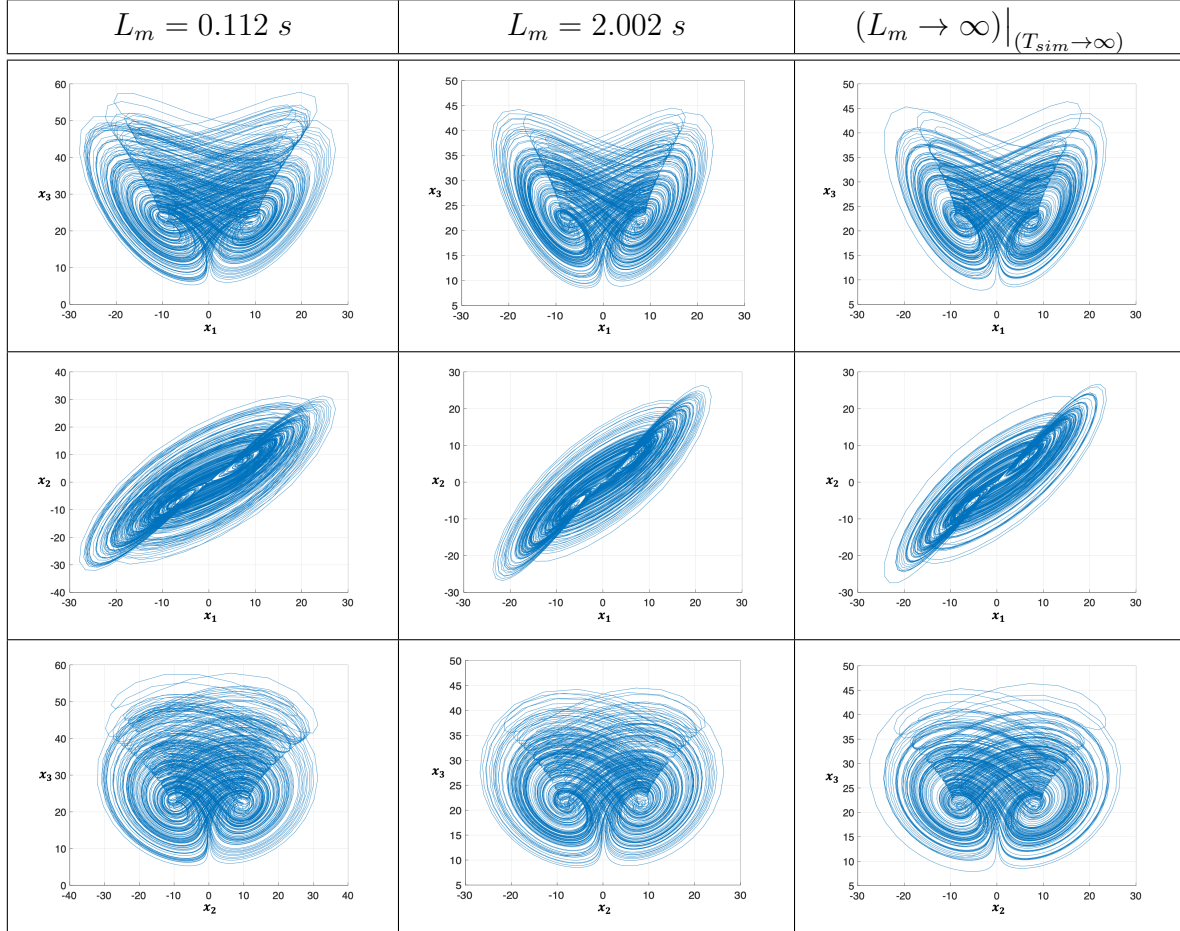
$$\begin{aligned}
C_0^q &= 1, \quad C_j^q = \left(1 - \frac{1+q}{j}\right) C_{j-1}^q, \quad j = 1, 2, \dots, \frac{L_m}{h}, \\
x_1(t_{k+1}) &= a(x_2(t_k) - x_1(t_k))h^q - \sum_{j=1}^v C_j^q x_1(t_{k-j}), \\
x_2(t_{k+1}) &= \left((c-a)(x_1(t_k) - x_1(t_k)x_3(t_k) + cx_2(t_k))\right)h^q - \sum_{j=1}^v C_j^q x_2(t_{k-j}), \\
x_3(t_{k+1}) &= (x_1(t_k)x_2(t_k) - bx_3(t_k))h^q - \sum_{j=1}^v C_j^q x_3(t_{k-j}),
\end{aligned} \tag{4.5}$$

con

$$v = \begin{cases} k & k < \frac{L_m}{h} \\ \frac{L_m}{h} & k \geq \frac{L_m}{h} \end{cases}.$$

En la tabla 4.1 se ilustra diferentes tipos de comportamiento caótico, obtenidos al variar el valor de L_m para el sistema de Chen de orden fraccional, considerando los parámetros, paso de integración y valores iniciales establecidos en la sección 2.1.2. En la primera columna, el atractor caótico es reconstruido considerando una longitud de memoria 0.112s, este valor es el mínimo requerido para garantizar convergencia al atractor caótico. En esta configuración la evolución del sistema presenta una convergencia bastante sensible a los parámetros establecidos, por lo tanto, una pequeña variación de estos valores tiene como consecuencia la destrucción del atractor caótico. En la segunda columna se considera una longitud de memoria de 2.002s. Aquí la convergencia del atractor caótico, presenta un comportamiento más robusto ante la variación en sus parámetros y condiciones iniciales, por lo tanto mantiene un mayor rango dinámico, lo que permite su aplicación en la realización experimental. Finalmente en la última columna se aprecia el atractor caótico utilizando la memoria completa del sistema.

Tabla 4.1: Resultados del principio de memoria corta en el método de Grunwald-Letnikov para el sistema de Chen de orden fraccional $q = 0.9$ a diferentes valores de L_m , con $(a, b, c) = (35, 3, 28)$, $T_{sim} = 90$ s, $h = 0.007$ y $(x(t_0), y(t_0), z(t_0)) = (1, 0, 25)$.



4.2.2. Oscilador caótico de Múltiples-Enrollamientos de orden fraccional basado en Grünwald-Letnikov

La solución numérica del sistema de Múltiples-Enrollamientos de orden fraccional, considerando el principio de memoria corta en el método de Grünwald-Letnikov, tiene la siguiente forma:

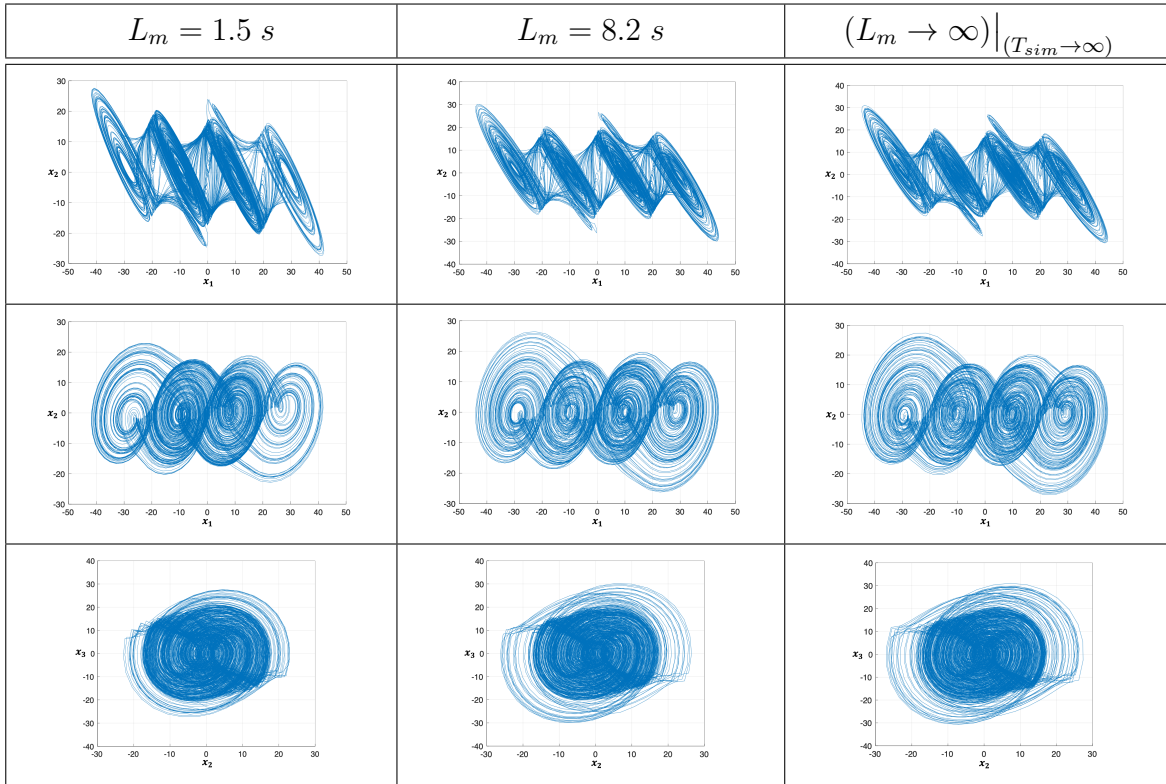
$$C_0^q = 1, \quad C_j^q = \left(1 - \frac{1+q}{j}\right) C_{j-1}^q, \quad j = 1, 2, \dots, \frac{L_m}{h},$$

$$\begin{aligned}
 x_1(t_{k+1}) &= x_2(t_k)h^q - \sum_{j=1}^v C_j^q x_1(t_{k-j}), \\
 x_2(t_{k+1}) &= x_3(t_k)h^q - \sum_{j=1}^v C_j^q x_2(t_{k-j}), \\
 x_3(t_{k+1}) &= \left(-ax_1(t_k) - bx_2(t_k) - cx_3(t_k) + df(x_1(t_k); k_s, h_s, p_s, q_s) \right) h^q - \sum_{j=1}^v C_j^q x_3(t_{k-j}),
 \end{aligned} \tag{4.6}$$

con

$$v = \begin{cases} k & k < \frac{L_m}{h} \\ \frac{L_m}{h} & k \geq \frac{L_m}{h} \end{cases}.$$

Tabla 4.2: Resultados del principio de memoria corta en el método de Grunwald-Letnikov para el sistema de Múltiples-Enrollamientos de orden fraccional $q = 0.9$ a diferentes valores de L_m , con $(a, b, c, d) = (2.2, 1.4, 0.6, 2.2)$, $T_{sim} = 2500$ s, $h = 0.15$ y $(x(t_0), y(t_0), z(t_0)) = (1, 0, 1)$



De igual forma, en la tabla 4.2 se ilustran los tipos de comportamiento caótico, obtenidos al variar el valor de L_m para el sistema de Chen de Múltiples-Enrollamientos de orden fraccional. En la primera columna, el atractor caótico es reconstruido conside-

rando una longitud de memoria mínima, si bien existen una convergencia a un atractor caótico para $L_m > 300 \mu s$, no es hasta que $L_m = 1.5 s$, cuando el atractor caótico tiene una convergencia similar al atractor original, no obstante se hace evidente que para este valor no es posible reconstruir todas las trayectorias. En la segunda columna se considera una longitud de memoria de $8.2 s$, bajo esta configuración es que el atractor caótico es considerado apto para su realización experimental. Finalmente en la última columna se aprecia el atractor caótico utilizando la memoria completa del sistema.

Cabe señalar que este tipo sistemas, requieren de una mayor longitud de memoria para preservar un mayor rango dinámico, esto se debe a que la evolución temporal requerida para formar el atractor caótico es mayor en comparación con otros sistemas caóticos. Otro factor que implica un mayor valor en la longitud de memoria L_m , es el valor del paso de integración h , no obstante, esto implica una mayor ventana de memoria $L = \frac{L_m}{h}$, es decir, un mayor número de elementos o de muestras embebidos, que termina compensando el incremento o decremento de la longitud de memoria. Como se ha explicado en la **sección 3.2**, desde el punto de vista computacional es recomendable utilizar la mínimo número de elementos para resolver el sistema.

4.3. Método 3: Métodos de Descomposición de Adomian para sistemas de orden fraccional

Como se ha abordado en el **Capítulo 1**, el Método de descomposición de Adomian (ADM) devuelve una solución analítica para cortos periodos de tiempo. Cuya complejidad recae en la descomposición en polinomios de Adomian de los tipos de no-linealidad presentes en varios sistemas dinámicos. No obstante, este método no considera funciones no lineales tales como: Heaviside, signo, valor absoluto, dientes de sierra, conmutación y funciones saturadas, las cuales son representados en varios sistemas dinámicos en la forma de funciones lineales por segmentos (PWL).

Considerando la motivación anterior, en esta sección describe la metodología para resolver sistemas de orden fraccional utilizando ADM y se propone un método basado en Adomian para resolver sistemas con funciones lineales por segmentos PWL. Este método se ha denominado Método de Descomposición de Adomian PWL, cuya principal característica es sustituir la descomposición en polinomios de Adomian, por un análisis de la soluciones obtenidas en cada región que compone una función PWL.

4.3.1. Método de Descomposición de Adomian

Considere el Método de Descomposición de Adomian (ADM)

$$\begin{aligned}
 \mathbf{x}^0(t_j) &= \frac{h^q}{\Gamma(q+1)} \mathbf{g}(t) + \mathbf{x}(t_0), \\
 \mathbf{x}^1(t_j) &= L\mathbf{x}^0(t_j) + \mathbf{A}^0(\mathbf{x}^0(t_j)), \\
 &\vdots \\
 \mathbf{x}^i(t_j) &= L\mathbf{x}^{i-1}(t_j) + \mathbf{A}^{i-1}(\mathbf{x}^0(t_j), \dots, \mathbf{x}^{i-1}(t_j)), \\
 \mathbf{x}(t_{j+1}) &= \sum_{k=0}^i \mathbf{x}^k(t_j) \frac{h^k q}{\Gamma(kq+1)},
 \end{aligned} \tag{4.7}$$

con $j = 1, 2, \dots, M$. Por conveniencia, los términos $\mathbf{x}^k(t)$ en (4.7) son expresados como un vector de coeficientes de Adomian $\mathbf{C}^k \in \mathbb{R}^n$, donde $\mathbf{C}^0 = \mathbf{x}^0(t_j)$, $\mathbf{C}^1 = \mathbf{x}^1(t_j)$ y así sucesivamente $\mathbf{C}^i = \mathbf{x}^i(t_j)$. De esta forma la expresión (4.7) es reescrita de la siguiente forma matricial:

$$\begin{bmatrix} x_1(t_{j+1}) \\ x_2(t_{j+1}) \\ \vdots \\ x_n(t_{j+1}) \end{bmatrix} = \begin{bmatrix} C_1^0 & C_1^1 & \dots & C_1^i \\ C_2^0 & C_2^1 & \dots & C_2^i \\ \vdots & \vdots & \ddots & \vdots \\ C_n^0 & C_n^1 & \dots & C_n^i \end{bmatrix} \begin{bmatrix} 1 \\ \frac{h^q}{\Gamma(q+1)} \\ \frac{h^{2q}}{\Gamma(2q+1)} \\ \vdots \\ \frac{h^{iq}}{\Gamma(iq+1)} \end{bmatrix}, \tag{4.8}$$

donde $\mathbf{C}^k \in \mathbb{R}^n$ ($k = 1, 2, \dots, i$), se define de la siguiente manera:

$$\begin{bmatrix} \mathbf{C}^0 \\ \mathbf{C}^1 \\ \mathbf{C}^2 \\ \vdots \\ \mathbf{C}^i \end{bmatrix} = \begin{bmatrix} \mathbf{g}(t) \frac{h^q}{\Gamma(q+1)} + \mathbf{x}(t_0) \\ \mathbf{C}^0 + \mathbf{A}^0(\mathbf{C}^0) \\ \mathbf{C}^1 + \mathbf{A}^1(\mathbf{C}^0, \mathbf{C}^1) \\ \vdots \\ \mathbf{C}^{i-1} + \mathbf{A}^{i-1}(\mathbf{C}^0, \dots, \mathbf{C}^{i-1}) \end{bmatrix}. \tag{4.9}$$

Existen dos esquemas principales para obtener y organizar los polinomios de Adomian. Como se observa en la expresión (2.40), el primer esquema consiste en derivar el operador no lineal $Nx(t)$ y evaluar este acorde al tipo de no linealidad. De acuerdo

con la referencia [66] , los primeros polinomios se determinan de la siguiente manera:

$$\begin{aligned}
 A^0 &= F(x^0), \\
 A^1 &= x^1 F'(x^0), \\
 A^2 &= x^2 F'(x^0) + \frac{1}{2i} (x^1)^2 F''(x^0), \\
 A^3 &= x^3 F'(x^0) + x^1 x^2 F''(x^0) + \frac{1}{3i} (x^1)^3 F'''(x^0), \\
 A^4 &= x^4 F'(x^0) + \left(\frac{1}{2i} (x^2)^2 + x^1 x^3 \right) F''(x^0) + \frac{1}{2i} (x^1)^2 x^2 F'''(x^0) \\
 &\quad + \frac{1}{4i} (x^1)^4 F^{iv}(x^0).
 \end{aligned} \tag{4.10}$$

El segundo esquema consiste en realizar una expansión en series del operador no lineal $Nx(t)$ de la siguiente forma:

$$F(x_1, x_1, \dots, x_n) = \left(\sum_{k=0}^{i-1} x_1^k \right)^{\gamma_1} \left(\sum_{k=0}^{i-1} x_2^k \right)^{\gamma_2}, \dots, \left(\sum_{k=0}^{i-1} x_{n-1}^k \right)^{\gamma_{n-1}} \left(\sum_{k=0}^{i-1} x_n^k \right)^{\gamma_n} \tag{4.11}$$

- Cabe señalar que la expansión anterior puede variar dependiendo del tipo de no linealidad del sistema; las expansiones en serie para la mayoría de no linealidades son mediante operaciones algebraicas, identidades trigonométricas o expansión en series de Taylor.
- La expansión en series obtenida por alguna técnica del punto anterior deben separarse apropiadamente para formar los polinomios de Adomian A^i , dos reglas son presentadas a continuación:
 - A^0 depende de la condición inicial x_0 , A^1 depende de x^0 , x^1 y así sucesivamente A^i depende de $x^0, x^1, \dots, x^i$.
 - La suma de los superíndices de los polinomios de Adomian para A^i debe ser igual a i , por lo tanto no todos los términos obtenidos de la expansión en series son tomados en cuenta para la aplicación del método.

Finalmente, el esquema metodológico para implementar sistemas caóticos de orden fraccional mediante el Método de Descomposición de Adomian se ilustra en la figura 4.5.

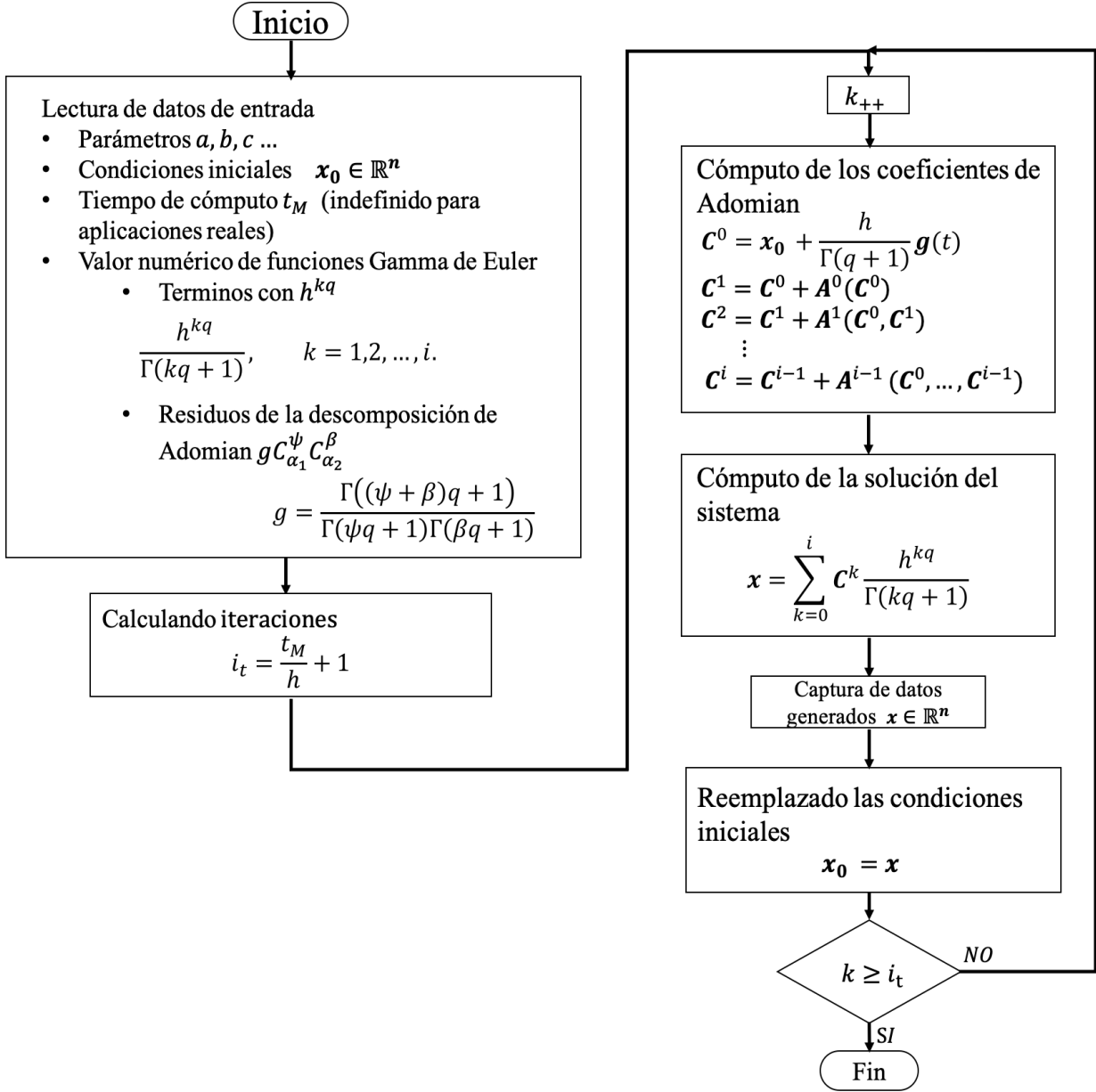


Figura 4.5: Diagrama de flujo para resolver sistemas caóticos en hardware embebido por el Método de Descomposición de Adomian.

Como se observa en el esquema anterior los términos que envuelven la función gamma de Euler $\frac{h^{kq}}{\Gamma(kq+1)}$ $k = 1, 2, \dots, i$, son previamente calculados para ahorrar tiempo

de cómputo. Estos valores son fijos y solo requieren ser calculados una vez durante todo el proceso iterativo. Otro tipo de términos que envuelven funciones gamma de Euler son los denominados residuos de la descomposición de Adomian, los cuales son obtenidos a partir de obtención de los polinomios de Adomian

$$\exists \underbrace{g C_{\alpha_1}^{\psi} C_{\alpha_2}^{\beta}} \in (A^0(C^0), A^1(C^0, C^1), \dots, A^{i-1}(C^0, \dots, C^{i-1})), \quad (4.12)$$

donde α_1, α_2 definen la membresía de variables de estado para los coeficientes, ψ y β definen los números de serie de la expansión y g es el residuo del método, determinado de la siguiente manera:

$$g = \frac{\Gamma[(\psi + \beta)q + 1]}{\Gamma(\psi q + 1)\Gamma(\beta q + 1)}. \quad (4.13)$$

En este método no existe un número fijo de funciones gamma de Euler, esto es debido por la expansión en series que devuelve un número no determinado de productos para cada no linealidad.

4.3.2. Método de Descomposición de Adomian PWL

Considere entonces que el término no lineal de la expresión (2.36) sea una función PWL $N\mathbf{x}(t) = f(x; \cdot)$, dada su naturaleza no puede ser expandida en polinomios de Adomian. Una forma general de aplicar la descomposición de Adomian para este tipo de no linealidad no se encuentra reportada. En la referencia [67] se propuso un método de interpolación polinomial para aproximar la combinación de una señal sinusoidal con una función PWL; con los polinomios interpolantes se realizó la expansión mediante método formal (2.38), no obstante, esta metodología devuelve un polinomio de grado superior, lo cual complica la expansión de Adomian dado el número de residuos que se generan (4.13). Lo que se propone en esta tesis, es una descomposición de Adomian basada en la construcción de modelos sobre diferentes regiones en la forma de sistemas lineales $\mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$. Este método propuesto se describe a continuación.

Por simplicidad, considere el siguiente sistema de orden fraccional

$${}^*D_{t_0}^q x_1(t) = ax_1 + f(x_1(t); m_H, b_H), \quad (4.14)$$

donde ${}^*D_{t_0}^q$ es el operador derivativo de Caputo, $a \in \mathbb{R}$ y

$$f(x_1(t); m_H, b_H) = \begin{cases} m_0 x_1(t) + b_0, & v_0, \\ m_1 x_1(t) + b_1, & v_1, \\ \vdots & \vdots \\ m_M x_1(t) + b_M, & v_M, \end{cases} \quad (4.15)$$

donde v_H y $m_H x + b_H$ con $H = 1, 2, \dots, M$ son las condiciones de membresía y ecuaciones de la recta respectivamente, estos elementos definen cada segmento lineal de la función PWL. Entonces, para una condición inicial $x_1(t_0)$, se procede a determinar la sección de membresía de la función

$$f(x_1(t_0); m_H, b_H) = m_H x_1(t_0) + b_H, \quad (4.16)$$

se sustituye el operador no lineal $N\mathbf{x}(t)$ de la expresión (2.37) por (4.16),

$$x_1(t) = x_1(t_0) + J_{t_0}^q L x_1(t) + J_{t_0}^q m_H x_1(t_0) + J_{t_0}^q g(t), \quad (4.17)$$

donde $g(t) = b_H$. Considerando la forma general de la solución en series infinitas

$$x(t) = \sum_{k=0}^i x^k(t), \quad (4.18)$$

los componentes x^k , $k = 1, 2, \dots, i$ para el sistema (4.14) son obtenidos mediante la siguiente forma recursiva, la cual se ha denominado método recursivo de Adomian PWL:

$$\begin{aligned} x_1^0(t) &= x_1(t_0) + J_{t_0}^q b_H, \\ x_1^1(t) &= J_{t_0}^q L x_1^0(t) + J_{t_0}^q m_H x_1^0(t), \\ x_1^2(t) &= J_{t_0}^q L x_1^1(t) + J_{t_0}^q m_H x_1^1(t), \\ &\vdots \\ x_1^i(t) &= J_{t_0}^q L x_1^{i-1}(t) + J_{t_0}^q m_H x_1^{i-1}(t). \end{aligned} \quad (4.19)$$

De forma análoga al Método de Adomian tradicional, la solución (4.18) tiene un radio de convergencia bastante pequeño, por tal motivo la trayectoria en un espacio de tiempo

$[t_0, t]$ se realiza en M subintervalos $[t_0, t_1], [t_1, t_2], \dots, [t_{M-1}, t_M]$, divididos por un tamaño de paso $h = (t_M - t_{M-1})$, y considerando que $C_1^0 = x_1^0(t_j)$, $C_1^1 = x_1^1(t_j)$ y sucesivamente $C_1^i = x_1^i(t_j)$, con $j = 1, 2, \dots, M$. La solución del sistema (4.14) toma la siguiente forma

$$\begin{aligned} C_1^0 &= x_1(t_0) + \frac{h^q}{\Gamma(q+1)} b_H, \\ C_1^1 &= LC_1^0 + m_H C_1^0, \\ C_1^2 &= LC_1^1 + m_H C_1^1, \\ &\vdots \\ C_1^i &= LC_1^{i-1} + m_H C_1^{i-1} \\ x_1(t_{j+1}) &= \sum_{k=0}^i C_1^k \frac{h^k q}{\Gamma(kq+1)} \quad k = 1, 2, \dots, i. \end{aligned} \tag{4.20}$$

Con base en la solución anterior, la técnica para generar atractores caóticos en orientaciones 1D, 2D y 3D presentada en la referencia [17] se puede adaptar de la siguiente forma:

$$\begin{aligned} \mathbf{C}^0 &= \mathbf{x}^0 + \frac{h^q}{\Gamma(q+1)} \mathbf{b}_{H_l} \\ \mathbf{C}^k &= \mathbf{A} \mathbf{C}^{k-1} + \mathbf{B} U(\mathbf{C}^{k-1}; \mathbf{m}_{H_l}), \quad l = 1, 2, \dots, n, \quad k = 1, 2, \dots, i, \end{aligned} \tag{4.21}$$

donde $\mathbf{x}^0$ es el vector de condiciones iniciales, $\mathbf{C}^k \in \mathbb{R}^n$ es el vector de coeficientes de Adomian que se desea calcular, $\mathbf{A} \in \mathbb{R}^{n \times n}$ es la matriz constante del sistema, $\mathbf{C}^{k-1} \in \mathbb{R}^n$ es el vector de coeficientes de Adomian calculados y $\mathbf{B} \in \mathbb{R}^{n \times n}$ es la matriz de coeficientes asociados al control PWL, $U(\mathbf{C}^{k-1}; \mathbf{m}_{H_l}) \in \mathbb{R}^n$, definido como:

$$U(\mathbf{C}^{k-1}; \mathbf{m}_{H_l}) = \begin{bmatrix} f_1(C_1^{k-1}; m_{H_1}) \\ f_2(C_2^{k-1}; m_{H_2}) \\ \vdots \\ f_n(C_n^{k-1}; m_{H_n}) \end{bmatrix}. \tag{4.22}$$

La representación matricial de (4.21) toma la siguiente forma:

$$\begin{aligned}
 \begin{bmatrix} C_1^1 \\ C_2^1 \\ \vdots \\ C_n^1 \end{bmatrix} &= \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix} \begin{bmatrix} C_1^0 \\ C_2^0 \\ \vdots \\ C_n^0 \end{bmatrix} + \begin{bmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,n} \\ b_{2,1} & b_{2,2} & \dots & b_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n,1} & b_{n,2} & \dots & b_{n,n} \end{bmatrix} \begin{bmatrix} f_1(C_1^0; m_{H_1}) \\ f_2(C_2^0; m_{H_2}) \\ \vdots \\ f_n(C_n^0; m_{H_n}) \end{bmatrix}, \\
 \begin{bmatrix} C_1^2 \\ C_2^2 \\ \vdots \\ C_n^2 \end{bmatrix} &= \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix} \begin{bmatrix} C_1^1 \\ C_2^1 \\ \vdots \\ C_n^1 \end{bmatrix} + \begin{bmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,n} \\ b_{2,1} & b_{2,2} & \dots & b_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n,1} & b_{n,2} & \dots & b_{n,n} \end{bmatrix} \begin{bmatrix} f_1(C_1^1; m_{H_1}) \\ f_2(C_2^1; m_{H_2}) \\ \vdots \\ f_n(C_n^1; m_{H_n}) \end{bmatrix}, \\
 &\vdots \\
 \begin{bmatrix} C_1^i \\ C_2^i \\ \vdots \\ C_n^i \end{bmatrix} &= \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix} \begin{bmatrix} C_1^{i-1} \\ C_2^{i-1} \\ \vdots \\ C_n^{i-1} \end{bmatrix} + \begin{bmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,n} \\ b_{2,1} & b_{2,2} & \dots & b_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n,1} & b_{n,2} & \dots & b_{n,n} \end{bmatrix} \begin{bmatrix} f_1(C_1^{i-1}; m_{H_1}) \\ f_2(C_2^{i-1}; m_{H_2}) \\ \vdots \\ f_n(C_n^{i-1}; m_{H_n}) \end{bmatrix},
 \end{aligned} \tag{4.23}$$

Finalmente la solución para un sistema de orden fraccional (2.35) basado en funciones PWL, tiene la siguiente forma matricial:

$$\begin{bmatrix} x_1(t_j + 1) \\ x_2(t_j + 1) \\ \vdots \\ x_n(t_j + 1) \end{bmatrix} = \begin{bmatrix} C_1^0 & C_1^1 & \dots & C_1^i \\ C_2^0 & C_2^1 & \dots & C_2^i \\ \vdots & \vdots & \ddots & \vdots \\ C_n^0 & C_n^1 & \dots & C_n^i \end{bmatrix} \begin{bmatrix} 1 \\ \frac{h^q}{\Gamma(q+1)} \\ \frac{h^{2q}}{\Gamma(2q+1)} \\ \vdots \\ \frac{h^{iq}}{\Gamma(iq+1)} \end{bmatrix}. \tag{4.24}$$

Cabe señalar que la solución de un sistema de orden fraccional obtenida mediante los métodos de Adomian y Adomian PWL dependen de un valor apropiado h para garantizar la convergencia, no obstante, dado que con Adomian PWL no se realiza una expansión en series, el método no genera productos entre coeficientes y por lo tanto el número de funciones Gamma de Euler de (4.24) siempre es igual a i , esto facilita las operaciones matemáticas, dado que el número de productos es menor.

Finalmente el esquema metodológico para implementar sistemas caóticos de orden fraccional mediante el Método de Descomposición de Adomian PWL se ilustra en la figura (4.6).

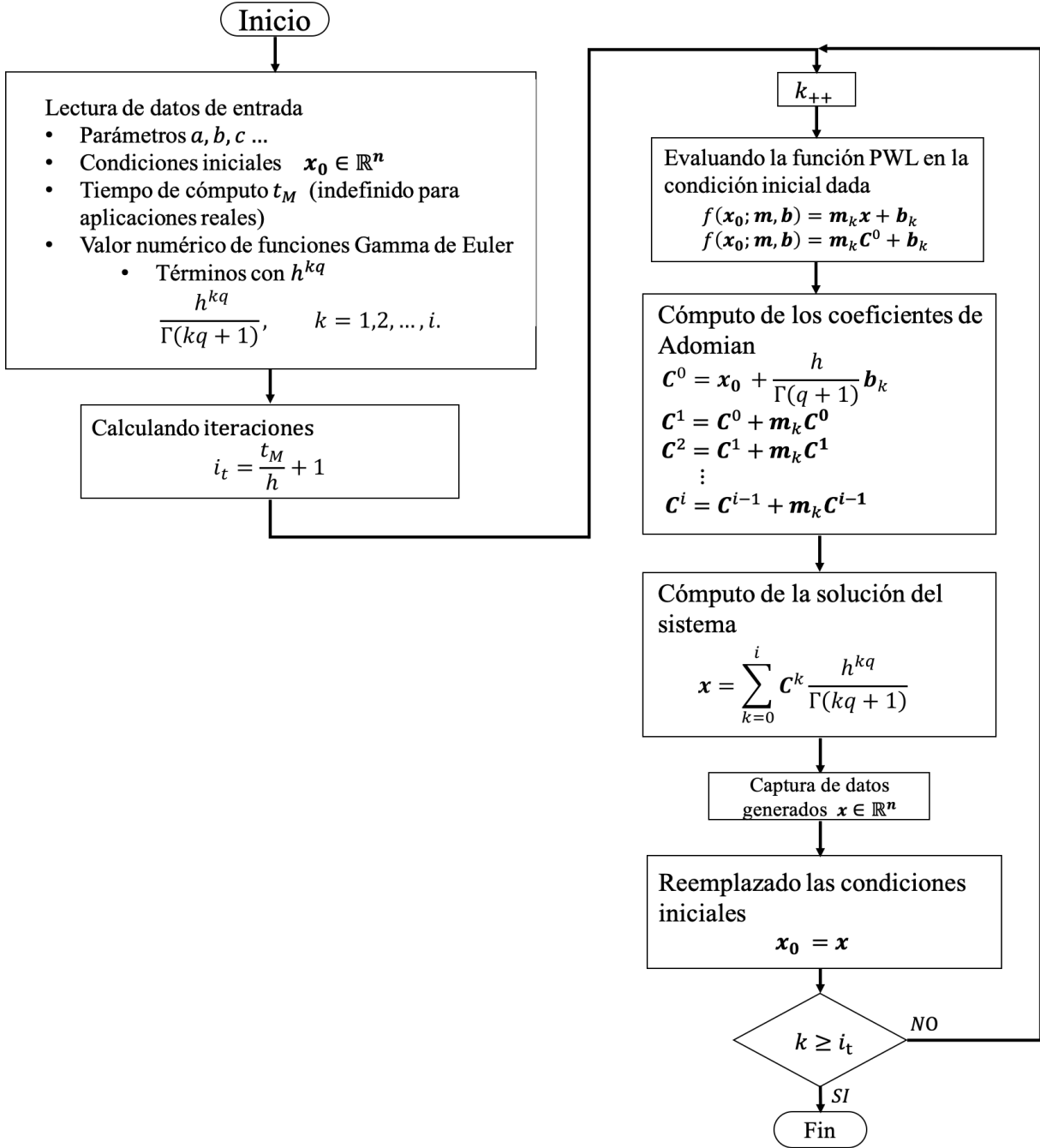


Figura 4.6: Diagrama de flujo para resolver sistemas caóticos en hardware embebido por el Método de Descomposición de Adomian PWL.

4.3.3. Oscilador caótico de Chen de orden fraccional basado en ADM

A continuación se presentan las ecuaciones que describen al sistema caótico de Chen de orden fraccional:

$$\begin{aligned} {}^*D_{t_0}^q x_1 &= a(x_2 - x_1), \\ {}^*D_{t_0}^q x_2 &= (c - a)x_1 - x_2x_3 + cx_2, \\ {}^*D_{t_0}^q x_3 &= x_1x_2 - bx_3, \end{aligned} \quad (4.25)$$

con parámetros $a = 35, b = 3, c = 28$.

Como se puede observar, el sistema de Chen no se encuentra definido mediante funciones PWL, por lo tanto se utiliza el método convencional de ADM para resolver este sistema:

Primero, se procede a calcular los respectivos polinomios de Adomian para los términos no lineales x_1x_3 y x_1x_2 . De acuerdo con (4.11), la expansión en series para los productos de variables de estado encontrados en (4.25), se realiza de la siguiente manera:

considere la función

$$F(x_1, x_2) = \left(\sum_{k=0}^{i-1} x_1^k \right)^{\gamma_1} \left(\sum_{k=0}^{i-1} x_2^k \right)^{\gamma_2}, \quad (4.26)$$

con $(\gamma_1 = \gamma_2 = 1)$. Considerando los primeros $i = 4$ términos de la expresión matricial (4.24), la expansión en series toma la siguiente forma

$$\begin{aligned} F(x_1, x_2) &= (x_1^0 + x_1^1 + x_1^2 + x_1^3)(x_2^0 + x_2^1 + x_2^2 + x_2^3), \\ &= x_1^0x_2^0 + x_1^0x_2^1 + x_1^1x_2^0 + x_1^0x_2^2 + x_1^1x_2^1 + x_1^2x_2^0 \\ &\quad + x_1^0x_2^3 + x_1^1x_2^2 + x_1^2x_2^1 + x_1^3x_2^0 + x_1^1x_2^3 \\ &\quad + x_1^2x_2^2 + x_1^3x_2^1 + x_1^2x_2^3 + x_1^3x_2^2 + x_1^3x_2^3. \end{aligned} \quad (4.27)$$

Posteriormente, se procede a separar los términos de (4.27) en los respectivos polinomios de Adomian

$$\begin{aligned} A^0 &= x_1^0x_2^0, \\ A^1 &= x_1^0x_2^1 + x_1^1x_2^0, \\ A^2 &= x_1^0x_2^2 + x_1^1x_2^1 + x_1^2x_2^0, \\ A^3 &= x_1^0x_2^3 + x_1^1x_2^2 + x_1^2x_2^1 + x_1^3x_2^0, \end{aligned} \quad (4.28)$$

se hace evidente que para

$$F(x_1, x_3) = \left(\sum_{k=0}^{i-1} x_1^k \right)^{\gamma_1} \left(\sum_{k=0}^{i-1} x_3^k \right)^{\gamma_3}, \quad (4.29)$$

con $(\gamma_1 = \gamma_3 = 1)$. Se descompone en los siguientes polinomios de Adomian

$$\begin{aligned} A^0 &= x_1^0 x_3^0, \\ A^1 &= x_1^0 x_3^1 + x_1^1 x_3^0, \\ A^2 &= x_1^0 x_3^2 + x_1^1 x_3^1 + x_1^2 x_3^0, \\ A^3 &= x_1^0 x_3^3 + x_1^1 x_3^2 + x_1^2 x_3^1 + x_1^3 x_3^0. \end{aligned} \quad (4.30)$$

De acuerdo con las expresión (4.9) y (4.13), la serie de coeficientes de Adomian $\mathbf{C}^k \in \mathbb{R}^3$ $k = 0, 1, \dots, 4$, para el sistema de Chen se definen de la siguiente manera:

$$\begin{aligned} \mathbf{C}^0 &\begin{cases} C_1^0 = x_1(t_0), & C_2^0 = x_2(t_0), & C_3^0 = x_3(t_0), \end{cases} \\ \mathbf{C}^1 &\begin{cases} C_1^1 = a(C_2^0 - C_1^0), \\ C_2^1 = (c - a)c_1^0 + cC_2^0 - C_1^0 C_3^0, \\ C_3^1 = -bC_3^0 + C_1^0 C_2^0, \end{cases} \\ \mathbf{C}^2 &\begin{cases} C_1^2 = a(C_2^1 - C_1^1), \\ C_2^2 = (c - a)c_1^1 + cC_2^1 - (C_1^0 C_3^1 + C_1^1 C_3^0), \\ C_3^2 = -bC_3^1 + C_1^0 C_2^1 + C_1^1 C_2^0, \end{cases} \\ \mathbf{C}^3 &\begin{cases} C_1^3 = a(C_2^2 - C_1^2), \\ C_2^3 = (c - a)c_1^2 + cC_2^2 - (C_1^0 C_3^2 + C_1^1 C_3^1 + g_1 C_1^1 C_3^1), \\ C_3^3 = -bC_3^2 + C_1^0 C_2^2 + C_1^1 C_2^1 + g_1 C_1^1 C_2^1, \end{cases} \\ \mathbf{C}^4 &\begin{cases} C_1^4 = a(C_2^3 - C_1^3), \\ C_2^4 = (c - a)c_1^3 + cC_2^3 - [C_1^0 C_3^3 + C_1^1 C_3^2 + g_2(C_1^1 C_3^2 + C_1^2 C_3^1)], \\ C_3^4 = -bC_3^3 + C_1^0 C_2^3 + C_1^1 C_2^2 + g_2(C_1^1 C_2^2 + C_1^2 C_2^1), \end{cases} \end{aligned} \quad (4.31)$$

donde

$$g_1 = \frac{\Gamma(2q + 1)}{\Gamma^2(q + 1)}, \quad g_2 = \frac{\Gamma(3q + 1)}{\Gamma(q + 1)\Gamma(2q + 1)}. \quad (4.32)$$

La solución numérica queda representada como

$$\begin{aligned} x_1(t_j + 1) &= C_1^0 + C_1^1 \frac{h^q}{\Gamma(q+1)} + C_1^2 \frac{h^{2q}}{\Gamma(2q+1)} + C_1^3 \frac{h^{3q}}{\Gamma(3q+1)} + C_1^4 \frac{h^{4q}}{\Gamma(4q+1)}, \\ x_2(t_j + 1) &= C_2^0 + C_2^1 \frac{h^q}{\Gamma(q+1)} + C_2^2 \frac{h^{2q}}{\Gamma(2q+1)} + C_2^3 \frac{h^{3q}}{\Gamma(3q+1)} + C_2^4 \frac{h^{4q}}{\Gamma(4q+1)}, \\ x_3(t_j + 1) &= C_3^0 + C_3^1 \frac{h^q}{\Gamma(q+1)} + C_3^2 \frac{h^{2q}}{\Gamma(2q+1)} + C_3^3 \frac{h^{3q}}{\Gamma(3q+1)} + C_3^4 \frac{h^{4q}}{\Gamma(4q+1)}. \end{aligned} \quad (4.33)$$

En la figura 4.7 se presenta los resultados de la simulación numérica a partir de la solución (4.33).

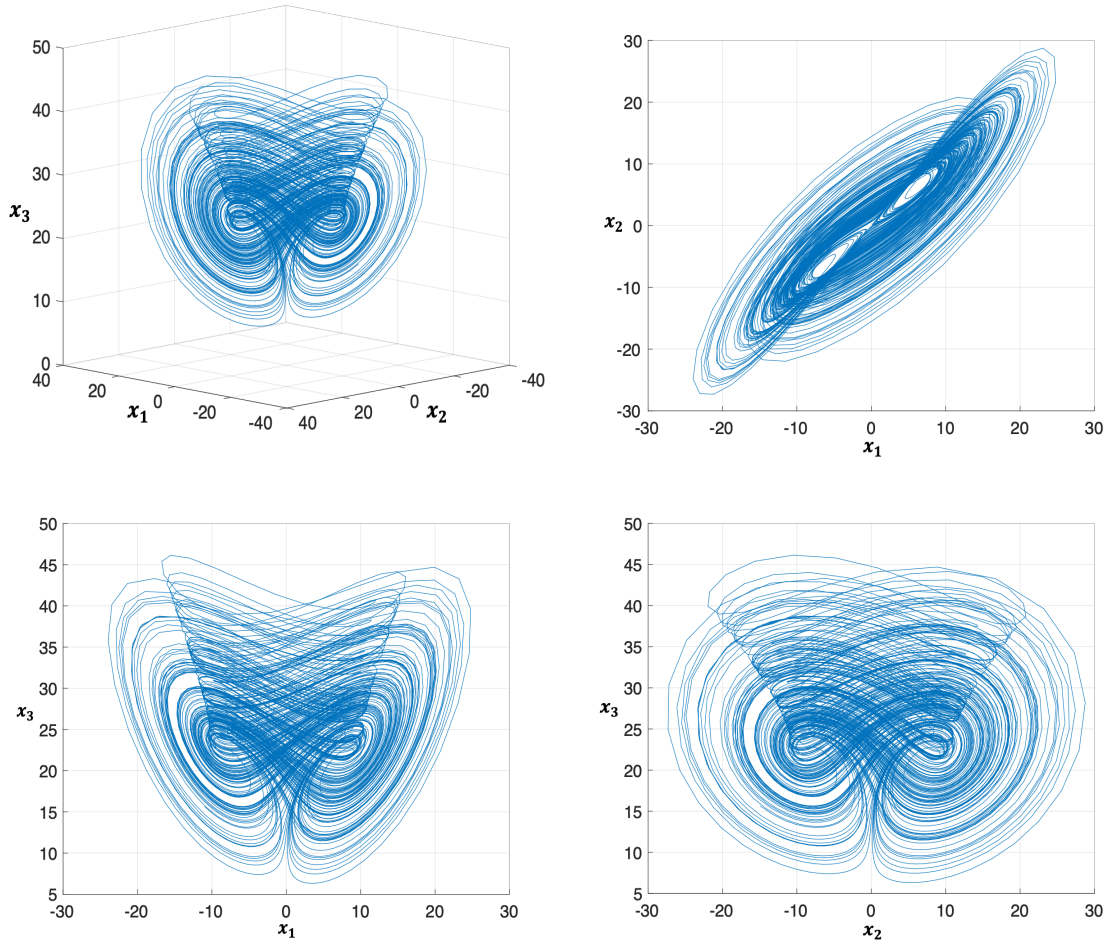


Figura 4.7: Resultado de la simulación del sistema de Chen de orden fraccional $q = 0.9$, realizada para $M = 10,001$ subintervalos, con $T_M = 60$ s, tamaño de paso $h = 0.006$ y condiciones iniciales $(x_1(t_0), x_2(t_0), x_3(t_0)) = (1, 0, 25)$.

4.3.4. Oscilador caótico de Múltiples-Enrollamientos de orden fraccional basado en ADM PWL

A continuación se presentan las ecuaciones que describen al sistema caótico de Múltiples-Enrollamientos:

$$\begin{aligned} {}^*D_{t_0}^q x_1 &= x_2, \\ {}^*D_{t_0}^q x_2 &= x_3, \\ {}^*D_{t_0}^q x_3 &= -ax_1 - bx_2 - cx_3 + df(x_1; k_s, h_s, p_s, q_s), \end{aligned} \quad (4.34)$$

con parámetros $a = 2.2$, $b = 1.4$, $c = 0.6$ y $d = 2.2$. Como se puede observar, el sistema de Múltiples Enrollamientos se encuentra definido por una serie de funciones saturadas, por lo tanto se utiliza el método propuesto de Adomian PWL para resolver este sistema. Al expandir las series de funciones saturadas $f(x_1; k_s, h_s, p_s, q_s)$ descrita en (3.21), se obtienen las condiciones para siete segmentos lineales $m_H x + b_H$, $H = 0, 1, 2, 3, 4, 5, 6$

$$f(x_1; k_s, h_s, p_s, q_s) = \begin{cases} m_0 x_1 + b_0, & x_s > q_s h_s + 1, \\ m_1 x_1 + b_1, & |x_s - i_1 h_s| \leq 1, -p_s \leq i_1 \leq q_s, \\ m_2 x_1 + b_2, & |x_s| \leq 1, -p_s \leq 0 \leq q_s, \\ m_3 x_1 + b_3, & |x_s - i_2 h_s| \leq 1, -p_s \leq i_2 \leq q_s, \\ m_4 x_1 + b_4, & i_1 h_s + 1 < x_s < (i_1 + 1)h_s - 1, -p_s \leq i_2 \leq q_s - 1, \\ m_5 x_1 + b_5, & i_2 h_s + 1 < x_s < (i_2 + 1)h_s - 1, -p_s \leq i_2 \leq q_s - 1, \\ m_6 x_1 + b_6, & x < -p_s h_s - 1, \end{cases} \quad (4.35)$$

donde

$$\begin{aligned} m_0 &= 0, & b_0 &= (2q_s + 1)k_s, \\ m_1 &= k_s, & b_1 &= -k_s i_1 h_s + 2i_1 k_s, \\ m_2 &= k_s, & b_2 &= 0, \\ m_3 &= k_s, & b_3 &= -k_s i_2 h_s + 2i_2 k_s, \\ m_4 &= 0, & b_4 &= (2i_1 + 1)k_s, \\ m_5 &= 0, & b_5 &= k_s, \\ m_6 &= 0, & b_6 &= -(2p_s + 1)k_s, \end{aligned} \quad (4.36)$$

con $i_1 = -1$ e $i_2 = 1$.

Ahora se procede a evaluar (4.35) en la condición inicial $x_1(t_0)$ para determinar su membresía, posteriormente se reemplaza la variable de estado por su respectivo coeficiente de Adomian:

$$\begin{aligned} f(x_1(t_0); k_s, h_s, p_s, q_s) &= m_H x_1 + b_H, \\ f(x_1(t_0); k_s, h_s, p_s, q_s) &= m_H C_1^0 + b_H, \end{aligned} \quad (4.37)$$

de acuerdo con las expresiones (4.23) y (4.24), las series de coeficientes de Adomian $C^k \in \mathbb{R}^3$ $k = 0, 1, \dots, 4$, para el sistema de Múltiples-Enrollamientos se definen de la siguiente manera

$$\begin{aligned} C^0 &\left\{ \begin{aligned} C_1^0 &= x_1(t_0), \quad C_2^0 = x_2(t_0), \quad C_3^0 = x_3(t_0) + \frac{h^q}{\Gamma(q+1)} b_H, \end{aligned} \right. \\ C^1 &\left\{ \begin{aligned} C_1^1 &= C_2^0, \\ C_2^1 &= C_3^0, \\ C_3^1 &= -aC_1^0 - bC_2^0 - cC_3^0 + dm_k C_1^0, \end{aligned} \right. \\ C^2 &\left\{ \begin{aligned} C_1^2 &= C_2^1, \\ C_2^2 &= C_3^1, \\ C_3^2 &= -aC_1^1 - bC_2^1 - cC_3^1 + dm_k C_1^1, \end{aligned} \right. \\ C^3 &\left\{ \begin{aligned} C_1^3 &= C_2^2, \\ C_2^3 &= C_3^2, \\ C_3^3 &= -aC_1^2 - bC_2^2 - cC_3^2 + dm_k C_1^2, \end{aligned} \right. \\ C^4 &\left\{ \begin{aligned} C_1^4 &= C_2^3, \\ C_2^4 &= C_3^3, \\ C_3^4 &= -aC_1^3 - bC_2^3 - cC_3^3 + dm_k C_1^3. \end{aligned} \right. \end{aligned} \quad (4.38)$$

La solución numérica queda representada de la siguiente manera:

$$\begin{aligned} x_1(t_j + 1) &= C_1^0 + C_1^1 \frac{h^q}{\Gamma(q+1)} + C_1^2 \frac{h^{2q}}{\Gamma(2q+1)} + C_1^3 \frac{h^{3q}}{\Gamma(3q+1)} + C_1^4 \frac{h^{4q}}{\Gamma(4q+1)}, \\ x_2(t_j + 1) &= C_2^0 + C_2^1 \frac{h^q}{\Gamma(q+1)} + C_2^2 \frac{h^{2q}}{\Gamma(2q+1)} + C_2^3 \frac{h^{3q}}{\Gamma(3q+1)} + C_2^4 \frac{h^{4q}}{\Gamma(4q+1)}, \\ x_3(t_j + 1) &= C_3^0 + C_3^1 \frac{h^q}{\Gamma(q+1)} + C_3^2 \frac{h^{2q}}{\Gamma(2q+1)} + C_3^3 \frac{h^{3q}}{\Gamma(3q+1)} + C_3^4 \frac{h^{4q}}{\Gamma(4q+1)}. \end{aligned} \quad (4.39)$$

En la figura 4.8 se presenta los resultados de la simulación numérica a partir de la solución (4.39).

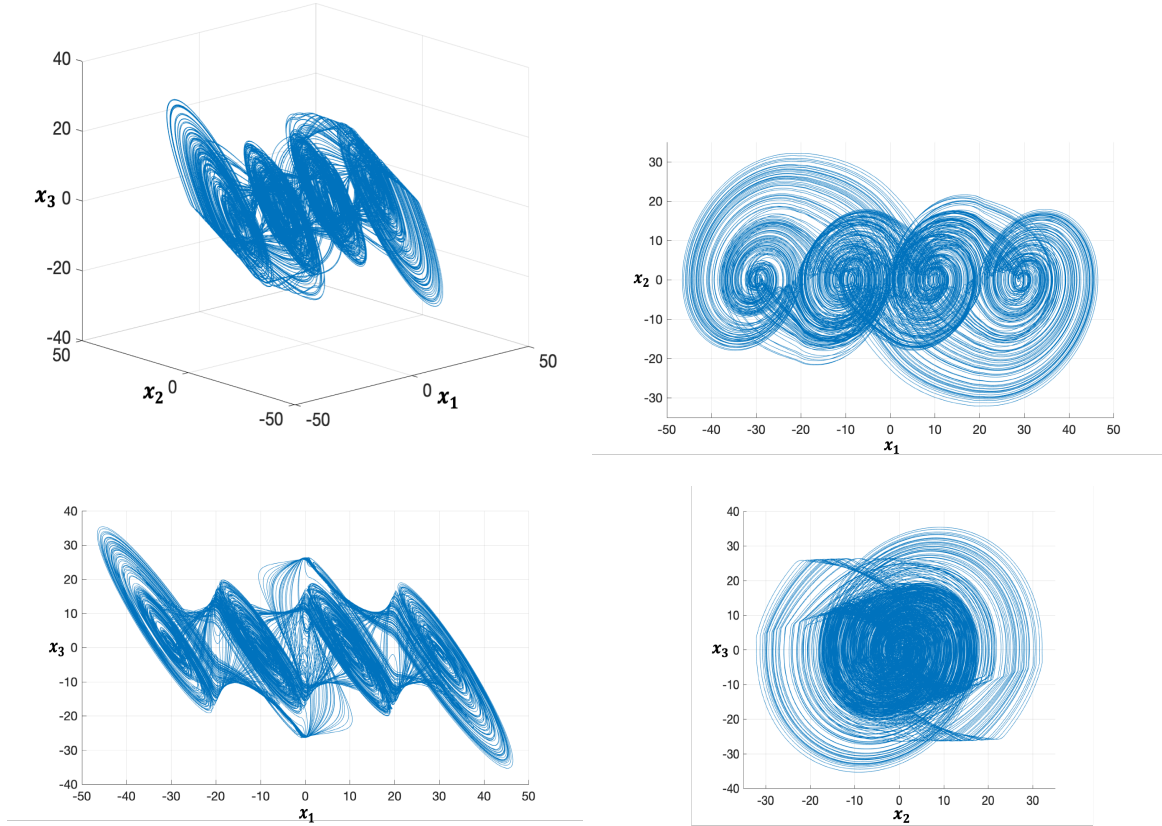


Figura 4.8: Resultado de la simulación del sistema de Múltiples-Enrollamientos de orden fraccional $q = 0.9$, realizada para $M = 200,001$ subintervalos, con $T_M = 2000$ s, tamaño de paso $h = 0.01$ y condiciones iniciales $(x_1(t_0), x_2(t_0), x_3(t_0)) = (1, 0, 1)$.

Capítulo 5

Realización de osciladores caóticos de orden entero y fraccional utilizando una plataforma ARM SoC

5.1. Configuración de la plataforma ARM SoC

El diagrama de bloques para la validación experimental de este trabajo se muestra en la figura 5.1. En la primera parte, el ARM SoC se encarga de resolver los modelos matemáticos correspondientes a los sistemas caóticos de orden entero y fraccional, en la segunda parte los datos obtenidos pasan por un proceso de conversión Digital-Analógico (D/A), finalmente los resultados obtenidos son capturados por un osciloscopio.

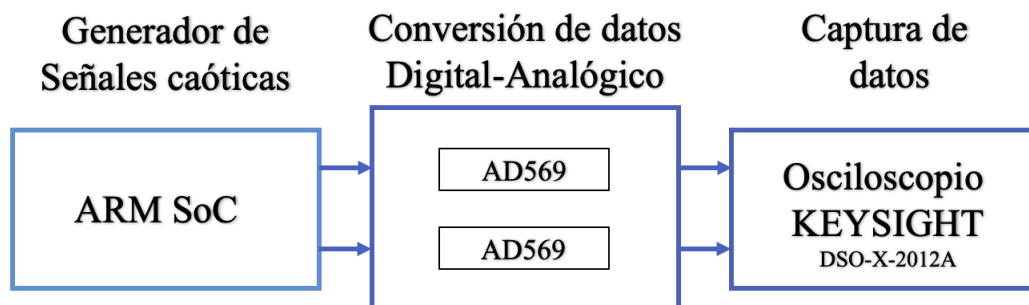


Figura 5.1: Diagrama de bloques para la validación experimental de osciladores caóticos de orden entero y fraccional.

El hardware experimental para la realización de osciladores caóticos de orden entero y fraccional, se muestra en la figura 5.2. La configuración de la tarjeta R-Pi 3 es similar al de una computadora convencional, por tal motivo se requieren periféricos de entrada y salida: ratón, teclado y una mini-pantalla HDMI para visualizar la interfaz gráfica. De igual forma, las tarjetas R-Pi requieren de una unidad de almacenamiento. Para este trabajo una tarjeta microSD de 8Gb es suficiente para guardar la configuración del software.

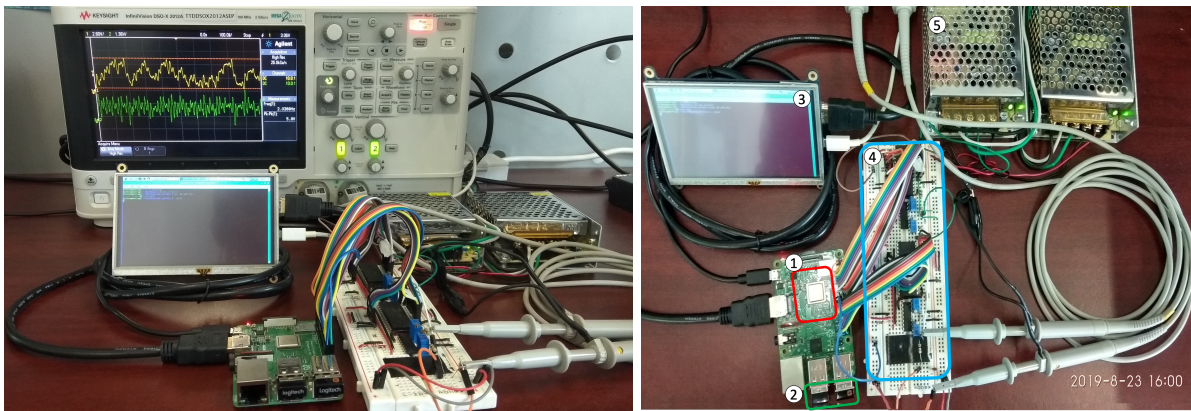


Figura 5.2: Hardware experimental utilizado para la implementación digital: 1) ARM SoC, 2) Periféricos USB para teclado y ratón, 3) Pantalla HDMI, 4) Convertidores D/A AD569 y 5) Fuente de alimentación.

5.1.1. Caracterización de los convertidores Digital-Analógico

La principal ventaja de las plataformas de desarrollo ARM SoC, frente a las computadoras de escritorio y portátiles, es la existencia del puerto GPIO (Entradas/Salidas de propósito general), con el cual es posible comunicar de forma directa y controlar componentes de hardware y microcontroladores. Para el caso de la tarjeta R-Pi 3, el puerto GPIO consiste de 40 pines de los cuales 28 son exclusivamente pines de salida/entrada digitales. En la figura 5.3 se ilustra la localización en el puerto GPIO de los pines digitales respecto a la numeración BCM (Broadcom SoC).

5 v	2	3 v	BCM 2
5 v	4	BCM 2	BCM 2
Gnd	6	BCM 3	BCM 3
BCM 14	8	BCM 4	BCM 4
BCM 15	10	Gnd	Gnd
BCM 18	12	BCM 17	BCM 17
Gnd	14	BCM 27	BCM 27
BCM 23	16	BCM 22	BCM 22
BCM 24	18	3.3 v	3.3 v
Gnd	20	BCM 10	BCM 10
BCM 25	22	BCM 9	BCM 9
BCM 8	24	BCM 11	BCM 11
BCM 7	26	Gnd	Gnd
BCM 1	28	BCM 0	BCM 0
Gnd	30	BCM 5	BCM 5
BCM 12	32	BCM 6	BCM 6
Gnd	34	BCM 13	BCM 13
BCM 16	36	BCM 19	BCM 19
BCM 20	38	BCM 26	BCM 26
BCM 21	40	Gnd	Gnd

Figura 5.3: Numeración de los pines de entrada/salida digitales del puerto GPIO.

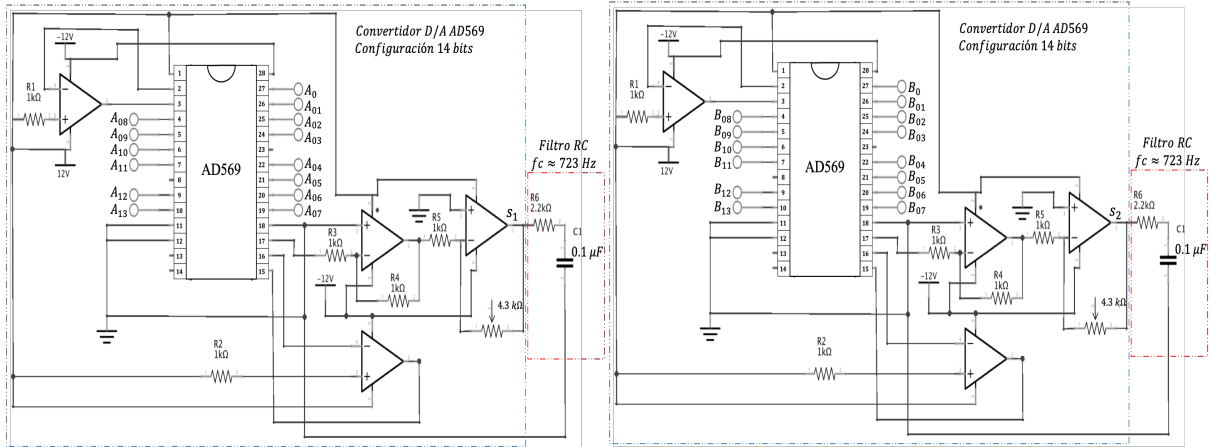


Figura 5.4: Diagrama esquemático para la conversión D/A. Primero, se definen dos canales A, B , a partir de estos se extraen los datos en formato Q_{14} (formato binario de 14 bits) de las señales que conforman los retratos de fase x_1x_3, x_1x_2 y x_2x_3 respectivamente. Después, cada canal es conectado a un convertidor D/A AD569. Finalmente, las señales analógicas S_1 y S_2 , son acondicionadas mediante un filtro RC.

La visualización de los datos generados por la realización experimental de osciladores caóticos, es un proceso que permite comparar y determinar de forma simple (en caso de existir) consistencia entre los resultados experimentales obtenidos por la metodología propuesta y resultados obtenidos por simulaciones numéricas. Para este propósito, considere el diagrama esquemático presentado en la figura 5.4. Cabe señalar que el convertidor AD569 tiene una resolución máxima de 16 bits, no obstante, este se ha configurado de acuerdo con los recursos disponibles de la tarjeta R-Pi 3. Por

otra parte, dado que la transferencia de bits entre estos dispositivos se realiza de forma serial, un filtro pasa-bajas es conectado a la salida de cada convertidor. El objetivo de este filtro es retener el valor real de la señal caótica para cada instante de tiempo t_k .

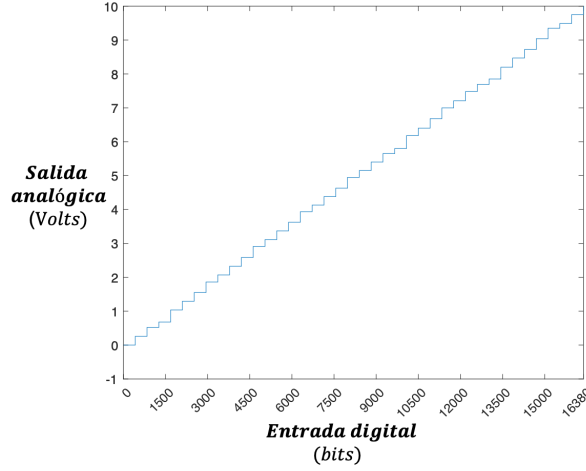


Figura 5.5: Respuesta típica de los convertidores D/A utilizados en el hardware experimental.

De forma ideal, un convertidor D/A está determinado por una función de transferencia lineal (salida analógica/entrada digital), no obstante, en la práctica se obtienen resultados no lineales asociados con errores de offset y ganancia intrínsecos a cualquier dispositivo analógico. Estos errores pueden ser expresados y corregidos usando

$$y = k(mx) + b - H, \quad (5.1)$$

donde y es la respuesta del convertidor, m es la pendiente que determina el error de ganancia, entonces k es la constante que ajusta esta pendiente; b es el error de offset y $H = b$ es una constante que compensa este error.

Por lo general el error de offset es un valor contante, y dada la configuración que se utiliza en esta tesis (figura 5.4), el compensador H puede implementarse fácilmente mediante la configuración del software, agregando el mismo valor a la entrada del DAC para el caso de offset negativo o añadiendo un restador a la salida del DAC para el caso de offset positivo. Por otra parte, el error de ganancia descrito por m presenta una mayor sensibilidad a las variaciones de temperatura del dispositivo y su ajuste matemático k requiere la implementación de circuitos electrónicos más complejos a los

cuales este trabajo de tesis no se encuentra orientado. Cabe señalar que para aplicaciones experimentales, donde el uso de convertidores D/A está orientado a la observabilidad de los datos y la precisión absoluta no es crítica, el error de ganancia es despreciable.

Considerando la explicación anterior, la respuesta típica de los convertidores D/A que se utilizan en esta tesis se observan en la figura 5.5, la gráfica ilustra la respuesta obtenida en volts en relación con la entrada digital $(0 - (2^{14} - 1))$. El error de offset para ambos convertidores es menor a $|5|mv$, por ser negativo, la constante de compensación H puede ser añadida mediante la configuración del software.

5.1.2. Interfaz gráfica

De forma análoga a las computadoras convencionales, las tarjetas R-Pi requieren de un sistema operativo por el cual acceder a las funciones del hardware, para este propósito, se instala **Raspbian** una distribución de Linux basada en Debian jessie con una interfaz gráfica de escritorio. Aunque en la actualidad existe un gran número de sistemas operativos que permiten ser cargados y utilizados por este tipo de hardware SoC, trabajar en un entorno de escritorio Linux simplifica la codificación, compilación y ejecución de programas de forma simple y eficiente. Por otra parte, el desarrollo de algoritmos dentro de esta plataforma se encuentra altamente orientada a lenguajes de alto nivel como Java o Python. No obstante, el uso de lenguajes de propósito general para aplicaciones basadas en hardware (concretamente el lenguaje de programación C), incrementan en rendimiento y permiten la portabilidad de estas aplicaciones (migración de los esquemas diseñados a otros ordenadores o plataformas ARM).

Considerando la motivación anterior, en la figura 5.6 se ilustra el desarrollo de un programa en lenguaje C dentro de un ambiente de gráfico Linux. Primero, un archivo de texto de extensión .C es requerido, en este archivo se desarrolla la estructura del código del programa, posteriormente el código es procesado por el compilador “gcc” (colección de compiladores GNU). Este compilador es nativo de las distribuciones de Linux y es fácilmente accesible desde la consola o terminal de Linux. De forma general, esta herramienta permite la interacción gráfica del usuario-máquina sin sacrificar un alto consumo de recursos computacionales.

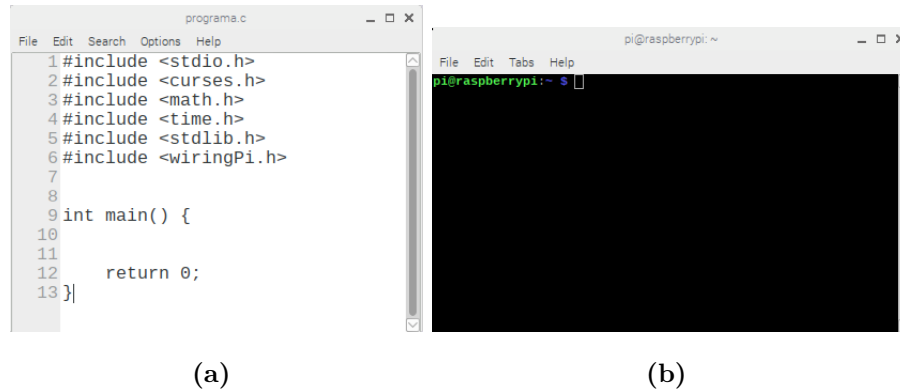


Figura 5.6: (a) Editor de texto para el desarrollo del programa y (b) Terminal de Linux

En la tabla 5.1 se presentan los comandos básicos para compilar, depurar y ejecutar el programa.

Comando	Descripción
<code>gcc programa.c -lm -o ejecutable -lwiringPi</code>	Compila el código en C y crea el archivo “ejecutable” correspondiente al programa diseñado.
<code>./ejecutable</code>	Inicia el programa.
<code>gcc -g programa.c -lm -o ejecutable -lwiringPi</code>	Incluye la información necesaria para depurar el programa.
<code>gdb ./ejecutable</code> Run	Corre el programa e informa el error del mismo.

Tabla 5.1: Comandos requeridos para el desarrollo de un programa en C dentro de la terminal de Linux.

Finalmente, en la figura 5.7 se ilustra el diagrama general para la realización experimental de osciladores caóticos de orden entero y fraccional. Primero, la tarjeta R-Pi 3 es inicializada, posteriormente el programa confirma los datos introducidos: parámetros, condiciones iniciales, etc. Durante el cómputo iterativo, la solución para un sistema caótico de orden entero o fraccional es obtenida mediante la respectiva metodología presentada en el **Capítulo 3**. Posteriormente, los datos generados para un tiempo t_k son enviados a través del puerto GPIO para su conversión D/A. Debido a que las variables de salida son digitales, una técnica de post-procesamiento es requerido para

redondear y compensar estos valores de salida, cabe señalar que esta técnica no afecta el cómputo de la solución debido a que no existe una retroalimentación de la salida a la entrada digital. Después de la conversión D/A un tiempo de espera es añadido, esto para variar el ancho de banda Bw del puerto GPIO, considerando el diagrama esquemático de la figura 5.4 se normaliza $Bw < f_c$. Finalmente las condiciones iniciales son remplazadas por los nuevos valores generados.

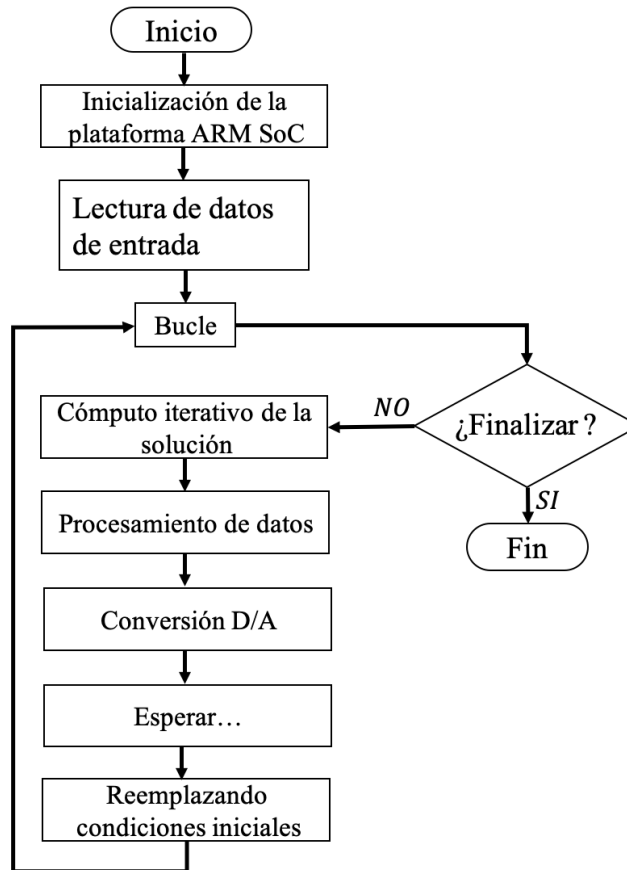


Figura 5.7: Diagrama de flujo de la implementación digital.

5.2. Resultados Experimentales

En esta sección se presentan los resultados obtenidos por la realización experimental de osciladores caóticos de orden entero y fraccional. La observación experimental se da de la siguiente manera: en las figuras 5.8-5.10 el oscilador caótico experimental de Chen es visualizado. De igual forma en las figuras 5.11-5.13 se presentan los resultados del oscilador Caótico experimental de Múltiples-Enrollamientos. Para el caso clásico, se

han implementado los modelos (3.10) y (3.32) basados en el método de Runge-Kutta de cuarto orden. Para el caso de orden fraccional se han implementado los modelos (4.5) y (4.6) basados en el principio de memoria corta del método de Grünwald-Letnikov y los modelos (4.31,4.38) basados en ADM y ADM PWL, respectivamente. Las figuras antes mencionadas ilustran las series de tiempo de las variables de estado x_1 , x_2 y x_3 y los respectivos retratos de fase del sistema. Cabe señalar que en la realización experimental se ha utilizado la misma configuración (parámetros, condiciones iniciales, etc.), utilizada para las simulaciones numéricas en MATLAB.

En los retratos de fase obtenidos es posible observar de forma clara el número de enrollamientos generados por cada oscilador, además, de las series de tiempo se puede inferir un comportamiento oscilatorio. De forma general, los resultados a nivel experimental muestran consistencia con los resultados obtenidos por simulaciones numérica, esto puede verificar la efectividad de la implementación de la metodología propuesta en este trabajo.

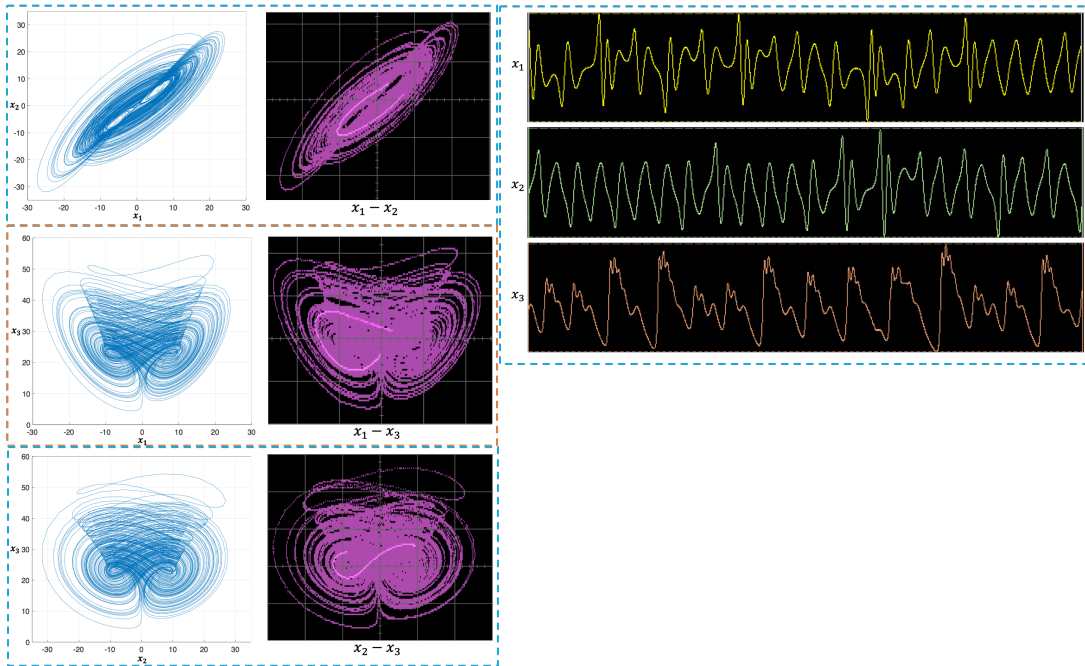


Figura 5.8: Resultados experimentales del oscilador caótico de Chen de orden entero basado en Runge-Kutta.

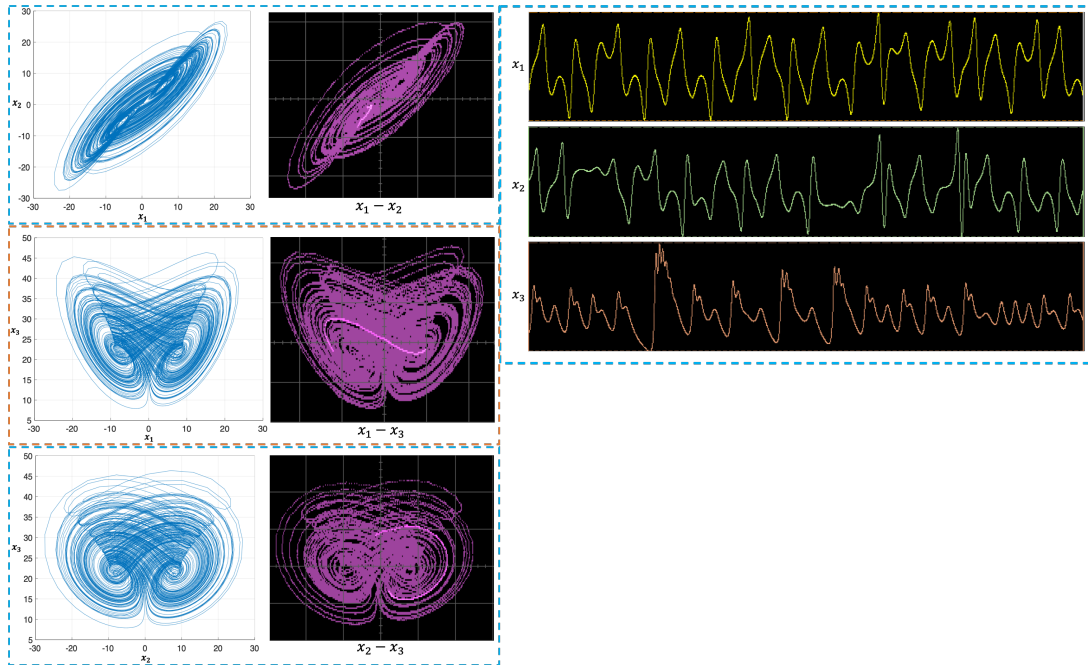


Figura 5.9: Resultados experimentales del oscilador caótico de Chen de orden fraccional basado en Grünwald-Letnikov.

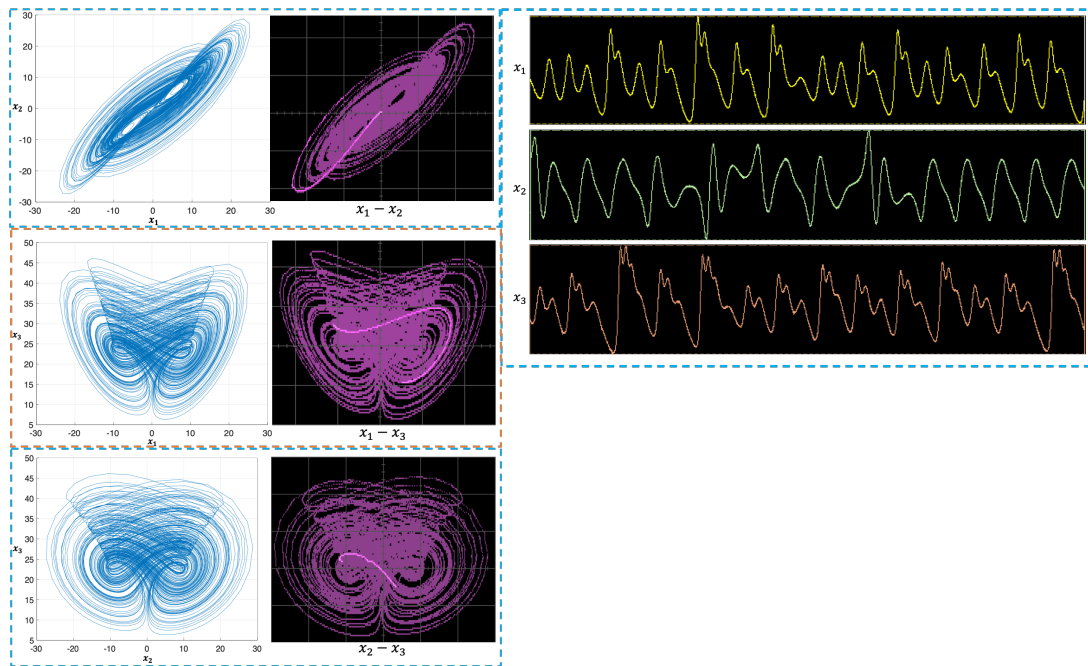


Figura 5.10: Resultados experimentales del oscilador caótico de Chen de orden fraccional basado en ADM.

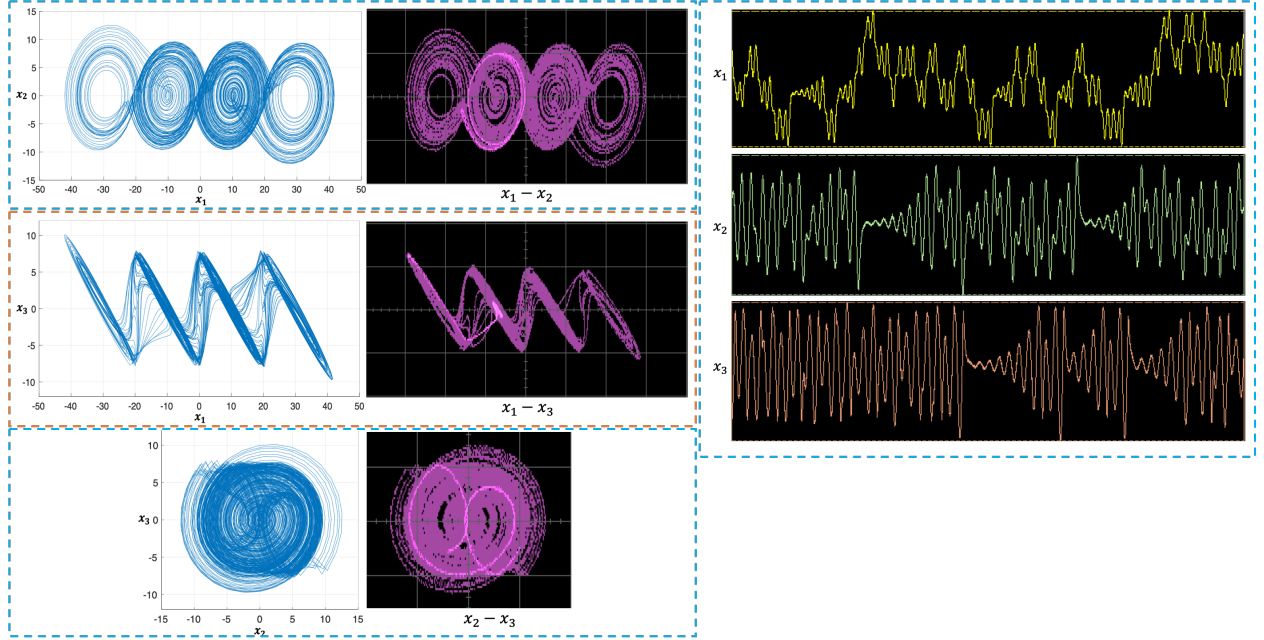


Figura 5.11: Resultados experimentales del oscilador caótico de Multiples-Enrollamientos de orden entero basado en Runge-Kutta.

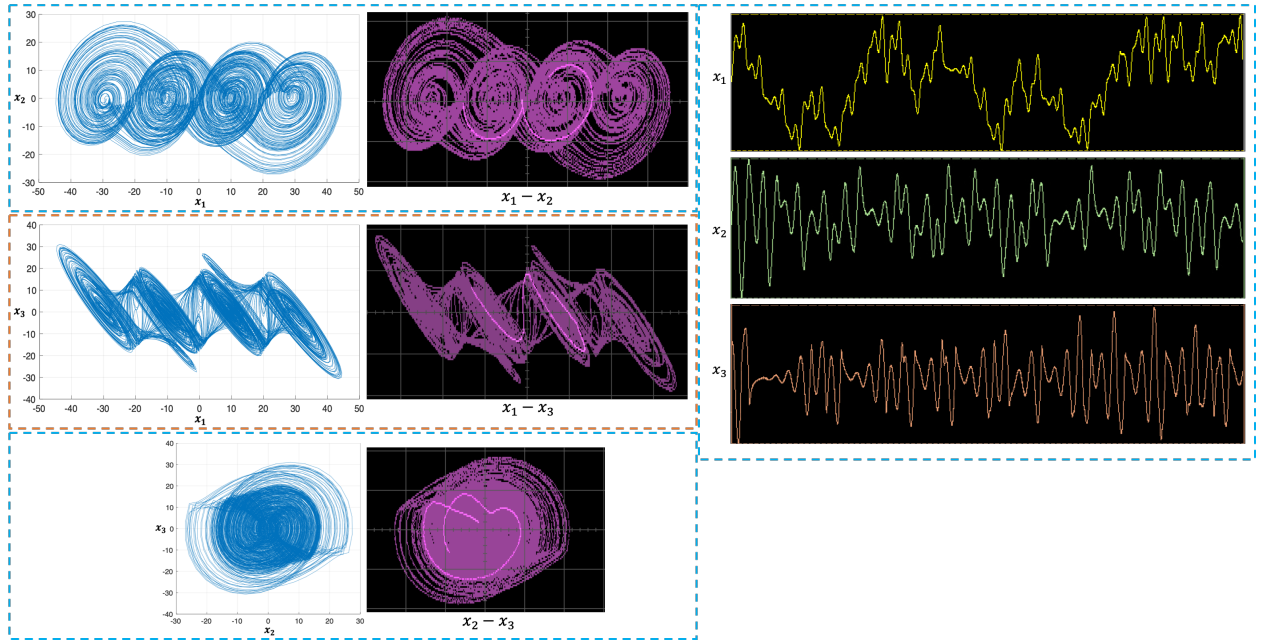


Figura 5.12: Resultados experimentales del oscilador caótico de Multiples-Enrollamientos de orden fraccional basado Grünwald-Letnikov.

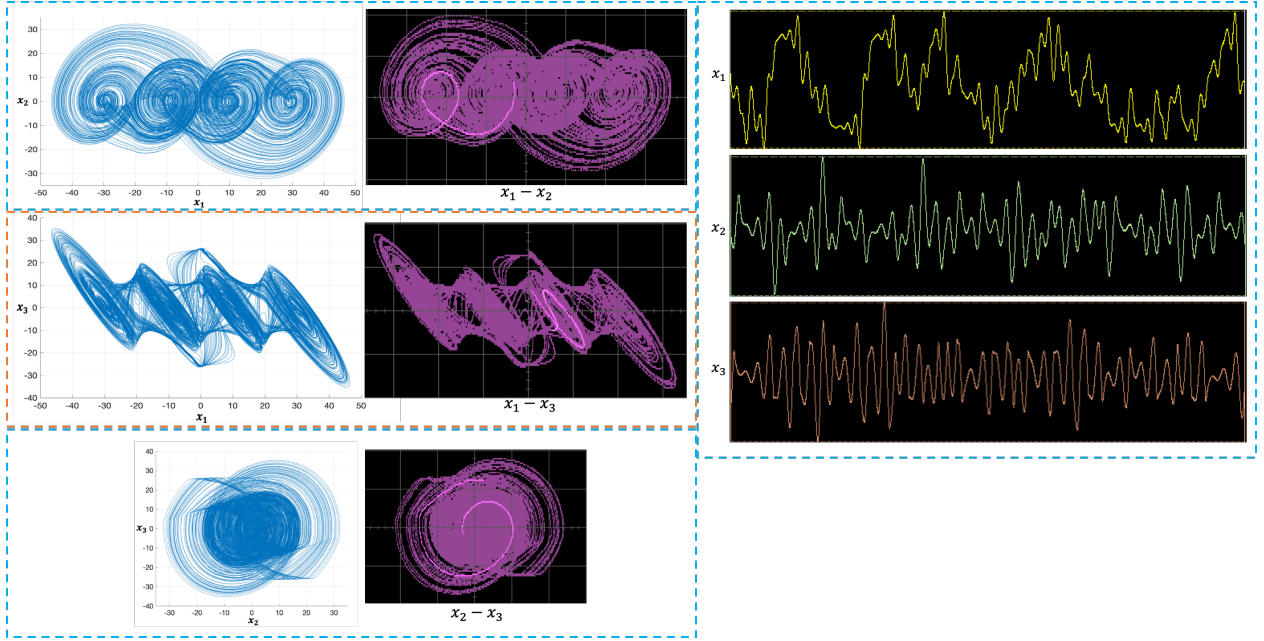


Figura 5.13: Resultados experimentales del oscilador caótico de Múltiples-Enrollamientos de orden fraccional basado en ADM PWL.

5.3. Validación del regimen caótico

Mediante los resultados obtenidos por la realización experimental de osciladores caóticos, es observable que las trayectorias obtenidas convergen a una forma similar de atractor obtenido por simulaciones numéricas, por esta razón es tentativo afirmar que un oscilador experimental mantiene un comportamiento caótico, no obstante, esta simple observación no afirma que para largos periodos de tiempo una trayectoria arbitraria pueda evolucionar desde un transiente caótico a un espacio restringido por curvas cerradas. Es por tal motivo, que en esta sección se realiza la validación del régimen caótico de cada oscilador caótico. Dicha validación es realizada a partir de las series de tiempo obtenidas de la realización experimental.

5.3.1. Mapas de Poincaré

Uno de los criterios para que un sistema dinámico presente un comportamiento caótico es la existencia de un conjunto denso de órbitas periódicas, en este sentido uno puede observar las consecuencias de estas órbitas mediante la construcción de mapas de Poincaré.

Considerando la motivación anterior, en la figura 5.14 se ilustra la manera en que se obtienen los mapas de Poincaré para los osciladores caóticos experimentales.

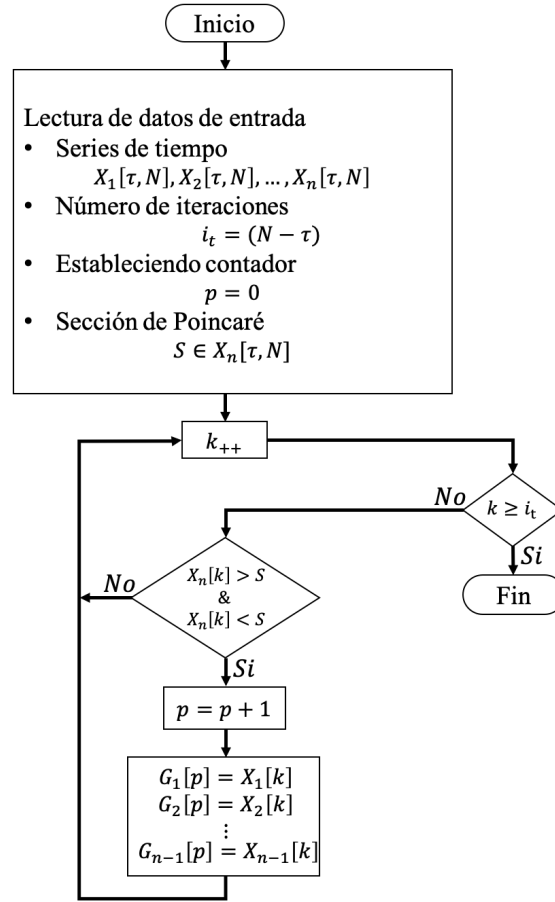


Figura 5.14: Diagrama de flujo para obtener el respectivo mapa de Poincaré de los osciladores caóticos experimentales.

En la tabla 5.2 se presenta la configuración utilizada en la generación de los mapas de Poincaré y el número intersecciones encontradas para un intervalo de tiempo $[\tau, N]$. Se ha considerado un valor de $\tau = 0.1N$, esto para eliminar posibles trayectorias que resulten en los mismos puntos de intersección obtenidos para los sucesivos puntos de partida de un estado transitorio $[0, (N - \tau)]$. Posteriormente la sección de Poincaré es asignada mediante el análisis previo de la geometría de cada atractor caótico.

Finalmente, en las figuras 5.15 y 5.16 se presentan los mapas de Poincaré de los osciladores experimentales, los cuales ilustran en un espacio de fase de dos dimensiones los puntos capturados durante la intersección de S .

Los resultados obtenidos por la cuantificación de los puntos de intersección está directamente relacionado con el conjunto de órbitas periódicas existentes, por otra parte, la densidad de estas órbitas es fácilmente observable en los mapas de Poincaré obtenidos, con lo cual se valida el comportamiento caótico de los osciladores experimentales.

Oscilador caótico experimental	Muestras (N)	Sección de Poincaré S	Número de intersecciones p
Chen de orden entero basado en Runge-Kutta	100,000	$S = 5.3 \in X_2[\tau, N]$	1247
Chen de orden fraccional basado en Grünwald-Letnikov	114,286	$S = 5 \in X_2[\tau, N]$	1158
Chen de orden fraccional basado en ADM	133,334	$S = 5 \in X_2[\tau, N]$	2040
Múltiples-Enrollamientos de orden entero basado en Runge-Kutta	20,000	$S = 4.3 \in X_3[\tau, N]$	692
Múltiples-Enrollamientos de orden fraccional basado en Grünwald-Letnikov	12,000	$S = 4 \in X_3[\tau, N]$	351
Múltiples-Enrollamientos de orden fraccional basado en ADM PWL	300,000	$S = 4 \in X_3[\tau, N]$	889

Tabla 5.2: Configuración y puntos de intersección encontrados para la generación de mapas de Poincaré de los osciladores caóticos experimentales.

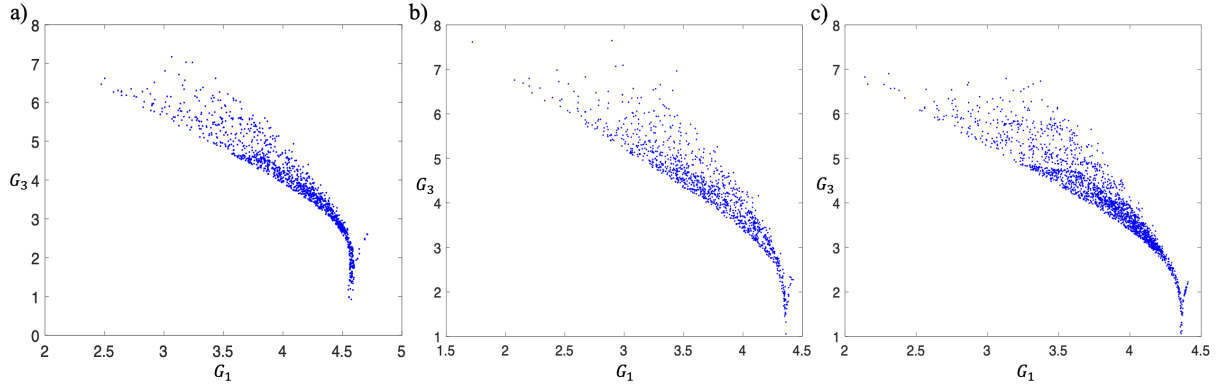


Figura 5.15: Mapas de Poincaré de los osciladores caóticos experimentales de Chen. a) Chen de orden entero basado en Runge-Kutta, b) Chen de orden fraccional basado en Grünwald-Letnikov y c) Chen de orden fraccional basado en ADM.

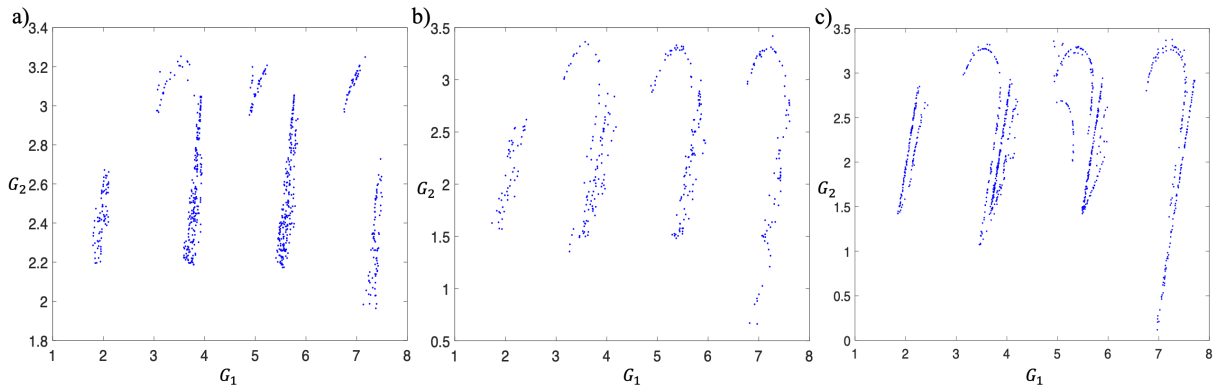


Figura 5.16: Mapas de Poincaré de los osciladores caóticos experimentales de Múltiples-Enrollamientos. a) Múltiples-Enrollamientos de orden entero basado en Runge-Kutta, b) Múltiples-Enrollamientos de orden fraccional basado en Grünwald-Letnikov y c) Múltiples-Enrollamientos de orden fraccional basado en ADM PWL.

5.3.2. Exponente máximo de Lyapunov

El análisis del exponente máximo de Lyapunov λ_{max} es una técnica fuertemente usada para determinar Caos, este cuantificador indica la separación exponencial en el tiempo de dos órbitas adyacentes en el espacio de fase con condiciones iniciales cercanas, y es una opción bastante útil cuando no se tiene la suficiente información sobre el sistema en cuestión. El cálculo de este exponente consiste en el análisis de las series de tiempo generadas por los osciladores caóticos experimentales (figuras 5.8-5.13) mediante el algoritmo de Rosenstein-Kantz. Para facilitar su aplicación se ha recurrido

a Tisean [68]. Este software dispone de librerías con métodos diseñados para el análisis de series temporales para sistemas dinámicos no lineales. Estas librerías son fácilmente implementables en lenguajes de alto nivel de programación desde C hasta MATLAB. En la tabla 5.3 se muestran los parámetros utilizados para la aplicación del algoritmo de Rosenstein-Kantz y los resultados obtenidos para el exponente máximo de Lyapunov de cada oscilador caótico.

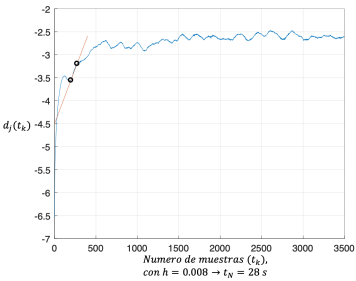
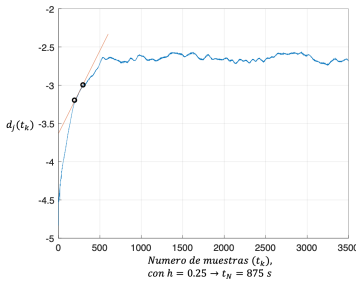
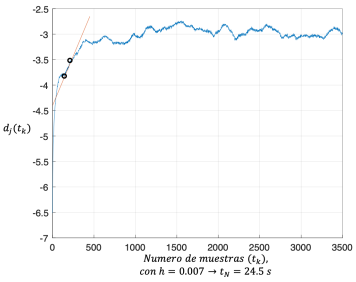
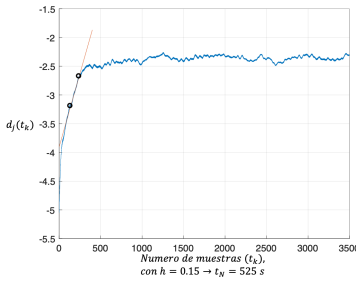
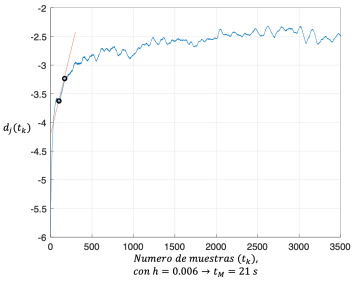
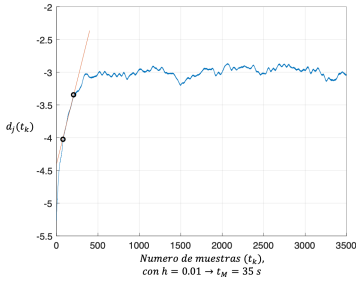
Oscilador caótico experimental	Dimensión (m)	Retardo (j)	Vecindad (min_X)	λ_{max}
Chen de orden entero basado en Runge-Kutta	3	2	0.07	0.4788
Chen de orden fraccional basado en Grünwald-Letnikov	3	2	0.09	0.3894
Chen de orden fraccional basado en ADM	3	4	0.09	0.5945
Múltiples-Enrollamientos de orden entero basado en Runge-Kutta	3	18	0.08	0.2169
Múltiples-Enrollamientos de orden fraccional basado en Grünwald-Letnikov	3	9	0.09	0.5090
Múltiples-Enrollamientos de orden fraccional basado en ADM PWL	3	12	0.06	0.5107

Tabla 5.3: Parámetros utilizados en el algoritmo de Rosenstein-Kantz para determinar el máximo exponente de Lyapunov.

Los resultados en la tabla anterior muestran que la existencia de un exponente

máximo de Lyapunov positivo $\lambda_{max} > 0$, para cada oscilador experimental, lo cual demuestra que las señales obtenidas por la realización experimental tienen un comportamiento caótico. Finalmente, en la tabla 5.4 se presentan las gráficas devueltas por el algoritmo: la función $d_j(t_k)$ contra el tiempo de reconstrucción t_N implícito por el número de muestras (t_k). Después de un periodo de transición, existe una región que se aproxima mediante una recta, el valor de la pendiente de dicha recta es igual al valor teórico $\lambda_{max} = \lim_{t \rightarrow \infty} \frac{1}{t} \ln \left| \frac{\delta x(t)}{\delta x(t_0)} \right|$, luego la curva logarítmica se satura dado que el sistema está acotado en el espacio y la tasa media de divergencia exponencial no puede exceder la región que ocupa el atractor en este espacio.

Tabla 5.4: Gráficas de la divergencia promedio de las trayectorias correspondientes a cada oscilador caótico.

	Chen	Múltiples-Enrollamientos
Osciladores caóticos experimentales de orden entero basados en Runge-Kutta		
Osciladores caóticos experimentales de orden fraccional basados en Grünwald-Letnikov		
Osciladores caóticos experimentales de orden fraccional basados en ADM y ADM PWL		

Capítulo 6

Conclusiones y Trabajo a futuro

En este trabajo de tesis se ha realizado el estudio teórico de los osciladores caóticos de Chen y de Múltiples-Enrollamientos, ambos en sus modelos de orden entero y fraccional. Las formulaciones teóricas se han validado numéricamente mediante simulaciones elaboradas en MATLAB. Para el caso de osciladores caóticos de orden entero, las simulaciones de los respectivos atractores caóticos se han realizado a partir de la solución numérica formulada por el método de Runge-Kutta de cuarto orden. Dicho método ha resultado factible para su implementación en hardware embebido.

Para el caso de osciladores caóticos de orden fraccional se ha utilizado el método de Grünwald-Letnikov para las simulaciones de los respectivos atractores caóticos, mediante este enfoque se analizó la propiedad de herencia o memoria de la derivada de orden fraccional, haciendo énfasis en la longitud de memoria como herramienta para analizar la derivada de una trayectoria de forma global y sus implicaciones en el consumo de recursos computacionales. Además, se estudio el concepto del principio de memoria corta en el método de Grünwald-Letnikov como alternativa para reducir el costo computacional en la obtención de las soluciones de osciladores caóticos de orden fraccional. No obstante, se detectó que esta técnica preserva elementos no requeridos en el cómputo numérico de la solución de un sistema, por tal motivo, se propuso una metodología para el manejo eficiente del principio de memoria corta en el método de Grünwald-Letnikov para la realización de osciladores caóticos experimentales. Además, se han realizado simulaciones con el objetivo de caracterizar la longitud de memoria adecuada para los sistemas de Chen y de Múltiples-Enrollamientos de orden fraccional.

Otro método propuesto para estudiar la dinámica caótica en orden fraccional, ha sido el método de descomposición de Adomian (ADM). Con este método se ha construido una solución cerrada semi-analítica para el oscilador caótico de Chen. Las funciones no lineales de este sistema se han aproximado mediante una expansión en series para formar los denominados “polinomios de Adomian”, con lo cual se eliminó la necesidad de establecer una longitud de memoria, dado que se observa la solución del sistema con una longitud de memoria completa para el número de expansiones en series realizadas. Cabe señalar que esto ha reducido el costo computacional considerablemente. Además se ha formulado un esquema metodológico para visualizar y facilitar la obtención de los denominados “residuos del método de Adomian”. Los resultados obtenidos por simulaciones son consistentes respecto a los resultados obtenidos por Grünwald-Letnikov.

Dada la importancia que tienen las funciones PWL para generar caos, se propuso un método de descomposición de Adomian lineal por segmentos, el cual es una extensión del método clásico de ADM para sistemas de orden fraccional. El concepto fundamental de este método ha sido el estudio cualitativo de las funciones PWL, con lo cual se ha eliminado la expansión en polinomios de Adomian y se ha reducido la complejidad del esquema de Adomian clásico. Adicionalmente se observó que ambos métodos convergen de forma rápida y permiten una implementación digital más conveniente dado que no requieren de un manejo complejo para el almacenamiento de datos.

Posteriormente, se llevó a cabo la realización experimental de osciladores caóticos en una tarjeta de bajo costo “Raspberry Pi 3 B+” acoplada a dos convertidores Digitales/Analógicos AD569, una caracterización de estos dispositivos fue realizada para obtener señales consistentes a partir de los datos de entrada digitales de los convertidores. Un enfoque abordado de este trabajo ha sido reducir los recursos de hardware que normalmente se utilizan en esquemas tradicionales de diseño digital de osciladores caóticos, por este motivo un factor discutido, fue el enfoque de programación. Dependiendo del paquete de software que se utilice la programación puede ser más sencilla, pero el script (código) debe abordar la recursividad y la creación de las variables intermedias de manera eficiente para evitar usar demasiada memoria, por este motivo se decidió utilizar el lenguaje de programación C, con una interfaz gráfica mínima para evitar una sobrecarga involucrada al crear los archivos necesarios de la realización experimental.

Los resultados experimentales capturados por un osciloscopio han sido consistentes

con los resultados obtenidos por simulaciones numéricas. Además, se han obtenido los mapas de Poincaré y exponentes máximos de Lyapunov a partir las series de tiempo generadas por cada oscilador caótico. Los resultados obtenidos por ambas técnicas validan que el comportamiento observado a nivel experimental es caótico. El trabajo aquí presentado es una base que permitirá el desarrollo de nuevos algoritmos para el análisis y solución de sistemas caóticos de orden entero y fraccional.

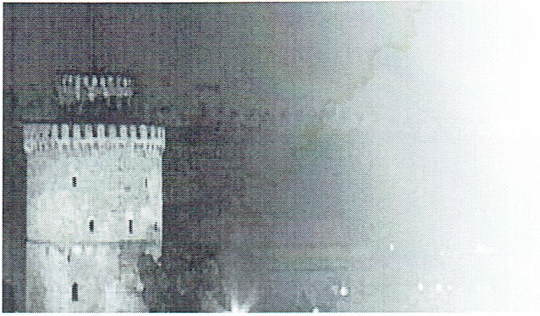
Como camino a seguir en trabajos futuros, se proponen los siguientes puntos:

- Implementar una arquitectura de diseño embebido en FPGA, para la realización experimental de osciladores caóticos mediante la metodología propuesta en este trabajo.
- Explorar aplicaciones de ingeniería basadas en caos para su realización experimental a partir de los osciladores caóticos diseñados.

Apéndice A. Participación en Congreso y Estancia Académica

En este apéndice se muestran las evidencias de la participación en el 8th congreso internacional Modern Circuits and Systems Technologies (MOCASST), realizado en la ciudad de Salónica, Grecia, del 13 al 15 de mayo de 2019. En este congreso se realizó la presentación del trabajo de investigación titulado “Efficient computation of the Grünwald-Letnikov for ARM-based implementations of Fractional-Order Chaotic Systems” en la modalidad de ponencia. Cabe señalar que en dicha participación, el trabajo quedó seleccionado como candidato para el premio al mejor papel de estudiante en Electrónica del congreso MOCASST 2019.

Por otro lado, se muestran las evidencias de una estancia académica realizada en el Departamento de física de la Universidad Aristóteles de Salónica, Salónica Grecia, del 01 de mayo al 30 de junio de 2019. En esta estancia se llevó a cabo el proyecto de investigación titulado “Implementación de sistemas caóticos de orden fraccional en plataformas FPGA”, bajo la asesoría y supervisión del Dr. Christos Volos, especialista en circuitos electrónicos no lineales.



MOCAST

INTERNATIONAL CONFERENCE ON
MODERN CIRCUITS AND SYSTEMS TECHNOLOGIES

This is to certify that

Clemente-lopez Daniel

attended the

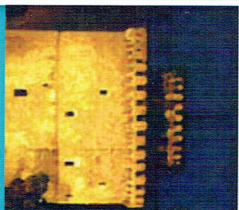
8th International Conference on Modern Circuits and System Technologies

held in Thessaloniki, Greece on May 13-15, 2019.

MOCAST 2019 General Chair

Prof. Spyridon Nikolaidis

MOCAST
INTERNATIONAL CONFERENCE ON
MODERN CIRCUITS AND SYSTEMS TECHNOLOGIES



MOCASST

INTERNATIONAL CONFERENCE ON
MODERN CIRCUITS AND SYSTEMS TECHNOLOGIES

13-15 May 2019, Thessaloniki, Greece

Nomination for the Best Student Paper and Presentation Award on Electronics

Presenter

Daniel Clemente-Lopez

Title

Efficient computation of the Grunwald-Letnikov method
for ARM-based Implementations of Fractional-Order Chaotic Systems

Prof. Spyridon Nikolaidis
Aristotle University of Thessaloniki
General Chair

Dr. Calliope-Louisa Sotiropoulou
Campera Electronic Systems Srl., Italy
Student Contest Chair

<http://mocast.physics.auth.gr>



Efficient computation of the Grünwald-Letnikov method for ARM-based Implementations of Fractional-Order Chaotic Systems

D. Clemente-López^{a*}, J. M. Muñoz-Pacheco^{b*}, O. G. Félix-Beltrán^{*}, C. Volos⁺

^{*}Faculty of Electronics Sciences, Autonomous University of Puebla, Puebla, México, C.P. 72570

⁺Aristotle University of Thessaloniki, Thessaloniki, GR, Greece C.P. 54124

^aironac77@gmail.com ^bjesusm.pacheco@correo.buap.mx

Abstract—In this work, an efficient scheme to compute the short memory principle of the Grünwald-Letnikov (GL) method is proposed. This scheme can be straightforwardly applied to implement fractional-order systems on processor-based hardware. To demonstrate its usefulness, a hyperchaotic fractional-order memristive system with hidden attractors is implemented on an ARM SoC low-cost board. The experimental phase portraits are consistent with the results obtained by numerical simulations.

Index Terms—Fractional-order, Grünwald-Letnikov, ARM, Hyperchaotic systems, Memristor, Hidden attractor.

I. INTRODUCTION

The application of fractional calculus to the analysis and implementation of chaotic systems has recently become a topic of interest in science and engineering [1]-[8]. Fractional-order derivatives preserve hereditary properties of diverse real phenomena, for instance, a fractional-order chaotic system with different families of self-excited and hidden attractors was introduced in [1] and a fractional-order financial system with negative parameters have been proposed in [2]. From the technical realization viewpoint, there are two common approaches to implement fractional-order chaotic systems using digital hardware [4]. By one hand, FPGA (Field Programmable Gate Array) realizations are a non embedded approach obtained with Verilog HDL [5]. On the other hand, an embedded approach oriented to the processors in ARM (advanced RISC machine)-based platforms is used [6].

In this work, we focus on implementations based on ARM platforms since the design is less complex than FPGAs implementations and the setup is performed by the Python or C programming languages [4]-[6]. Also, they are useful for Ethernet transmission. Therefore, ARM-based realizations of fractional-order chaotic systems could be more suitable for increasing security in novel applications, such as the Internet of Things (IoT), sensor networks, etc. because the ARM architectures are extensively used in consumer electronic devices (smart-phones, tablets, and other mobile devices).

However, the main issue of the ARM-based implementations is related to the complexity of the numerical algorithm used to discretize the fractional-order dynamical system. In literature, a good proposal oriented to minimize the computational effort consists of applying the Grünwald-Letnikov method which incorporates the short-memory principle [7].

It permits to select a memory length (L_m) to approximate the fractional-order. A proper selection of L_m is crucial to obtain accurate results of chaotic behaviors as was shown in [5]. Nevertheless, due to the Grünwald-Letnikov method was conceived as a numerical algorithm and ARM platforms are based on processors, the data generated by this algorithm for all simulation time are not deleted causing an accumulation of stored data in the RAM memory of the hardware [4]-[6].

In this manner, we introduce an efficient computational scheme to compute the short-memory principle of the Grünwald-Letnikov method in digital hardware. We have proved the proposed method by implementing a hyperchaotic fractional-order memristive system with hidden attractors on an ARM SoC low-cost board. Additionally, the scheme can be straightforwardly applied to implement fractional-order systems on processor-based hardware without loss of generality.

The paper is organized as follows. Section II gives the mathematical background of the Grünwald-Letnikov method. Section III introduces the proposed efficient scheme for the short-memory principle. Section IV describes the hyperchaotic fractional-order memristive system with hidden attractors. Section V shows the experimental results on embedded hardware. Finally, the conclusion is summarized in Section VI.

II. GRÜNWALD-LETNIKOV METHOD

According to the explicit numerical approximation of Grünwald-Letnikov, the fractional-order derivative q for a discrete function $f(t_k)$ can be described as follows [2]:

$${}_{k-\frac{L_m}{h}}D_{t_k}^q f(t_k) \approx h^{-q} \sum_{j=0}^k C_j^q f(t_{k-j}), \quad (1)$$

where L_m is the memory length, h is the integration step size, and C_j^q are the binomial coefficients which are calculated recursively with

$$C_0^q = 1, \quad C_j^q = \left(1 - \frac{1+q}{j}\right) C_{j-1}^q, \quad j = 1, 2, \dots, k. \quad (2)$$

Then, the numerical solution for a fractional-order differential equation (FODE) expressed in the form

$$D_{t_k}^q x(t_k) = f(x(t_k)), \quad (3)$$

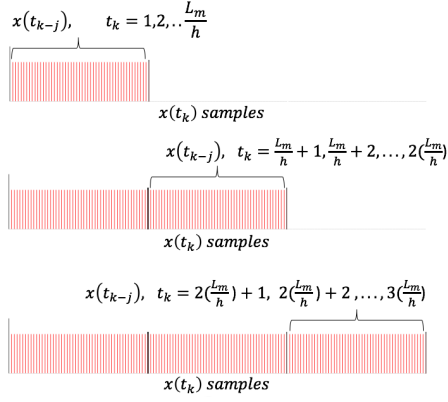


Fig. 1. A usual numerical approach for solving FODEs with the Grünwald-Letnikov method (4). The parentheses denote just the samples used to compute the short-memory principle in (5), however, the preceding samples are stored for all simulation time.

can be obtained by

$$x(t_k) = f(x(t_{k-1}))h^q - \sum_{j=1}^k C_j^q x(t_{k-j}). \quad (4)$$

for $0 < q < 1$. Expression (4) requires infinity memory length for numerical solution since the summation term depends on the discretized time t_k . This implies relatively high simulation times.

To reduce the computational time, the upper bound of summation needs to be modified by $k = v$, where

$$v = \begin{cases} k, & k \leq M, \\ L_m, & k > M. \end{cases} \quad (5)$$

This is known as the *short-memory principle*, where M is the memory window with a width defined by $M = \frac{L_m}{h}$. As was reported in [5], the accuracy increases by augmenting the width of memory window.

III. EFFICIENT COMPUTATION OF THE SHORT-MEMORY PRINCIPLE IN GRÜNWALD-LETNIKOV METHOD

An crucial aspect for digital platforms based on processors is the computational resources utilized by the computation of a program. By analyzing expression (4), we observe that the summation term can be represented by a dot product operation as follows

$$[C_1^q \ C_2^q \ C_3^q \ \dots \ C_v^q] \begin{bmatrix} x(t_{k-1}) \\ x(t_{k-2}) \\ x(t_{k-3}) \\ \vdots \\ x(t_{k-v}) \end{bmatrix}. \quad (6)$$

where $C^q \in \mathbb{R}^{1 \times v}$ is a binomial coefficients vector with fixed dimension, and $x(t_{k-v}) \in \mathbb{R}^{v \times 1}$ represents the vector of the preceding solutions. However, t_k does not have a fixed value, and hence $x(t_{k-1})$ needs to be updated after each iteration

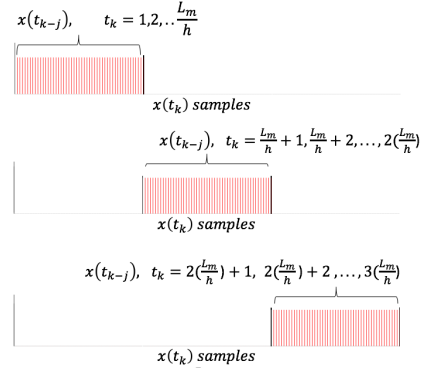


Fig. 2. The proposed approach to compute the short-memory principle where the previous samples are omitted. The parenthesis denotes the $x(t_k)$ samples used in vector $x(t_{k-1})$ as the number of iterations increases.

t_{k+1} . From a computational viewpoint, the Grünwald-Letnikov method (4) with the short-memory principle (5) does not remove the preceding solutions $x(t_k)$, which have been already utilized. Accordingly, we obtain a vector of the solutions $x(t_k) \in \mathbb{R}^n$ with a high dimension as the number of iterations tends to infinity. Figure 1 depicts this issue.

Nevertheless, the digital platforms based on processors (ARM boards) have limited hardware resources, such as the computational memory dedicated to storing data. Besides, a large number of stored data may reduce the performance of the computation as time increases. Therefore, the expressions (4) and (5) are not still suitable for digital implementation. In this manner, we introduce an efficient computation of the short-memory principle. More specifically, the proposed approach deletes the preceding samples and only preserves the last samples of the vector $x(t_k)$ as shown in Fig. 2. As a consequence, the stored data are limited by the memory window $M = \frac{L_m}{h}$. Due to that engineering applications based on chaos regularly requires just a few samples of the solution as in random number generators, the deleted samples of the solution do not corrupt the chaotic behavior. It is obvious that this approach cannot reconstruct the whole chaotic attractor for reduced sizes of M . To carry out the proposed approach, we propose the algorithm presented in Fig. 3.

IV. 4-D HYPERCHAOTIC FRACTIONAL-ORDER MEMRISTIVE SYSTEM WITH HIDDEN ATTRACTORS

In this section, the memristive system we proposed for the digital implementation on an ARM board is given. Recently, Volos et al. [8], have proposed a novel 4D-fractional-order memristive system, which is defined by

$$\begin{aligned} D_t^q x &= -10x - bu - uz, \\ D_t^q u &= -6x + 1.2xz + 0.1y(u, w) - c, \\ D_t^q z &= -z - 1.2u^2, \\ D_t^q w &= u, \end{aligned} \quad (7)$$

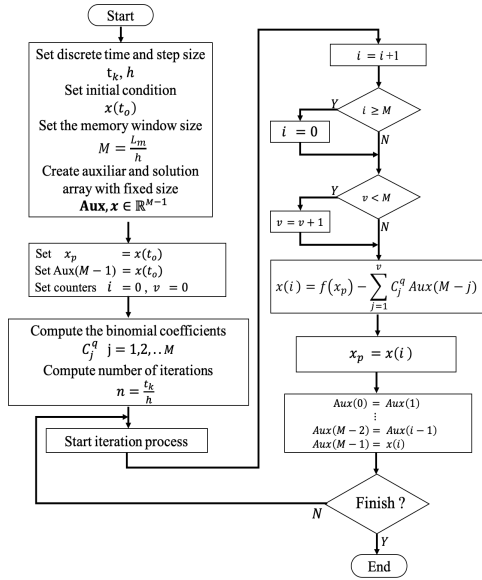


Fig. 3. Proposed algorithm for an efficient implementation of the short memory principle in (4). The solution vector $\mathbf{x}(t_k)$ has fixed dimension whereas an auxiliary vector $\mathbf{Aux}$ is generated with the previous solution vector.

where b and c are positive constant, q is the fractional-order, and $y(u, w)$ is the memristive device described by

$$y(u, w) = (1 + 0.24w^2 - 0.0016w^4)u. \quad (8)$$

As well known, equation (8) describes the change of its resistance by the charge that flows through it. The numerical solution of fractional-order memristive system (7) is obtained by applying the Grünwald-Letnikov method (4) as follows

$$\begin{aligned} x(t_k) &= f_1(x(t_{k-1}), u(t_{k-1}), z(t_{k-1}))h^q - \sum_{j=1}^k C_j^q x(t_{k-j}), \\ u(t_k) &= f_2(x(t_{k-1}), u(t_{k-1}), z(t_{k-1}), w(t_{k-1}))h^q - \sum_{j=1}^k C_j^q u(t_{k-j}), \\ z(t_k) &= f_3(u(t_{k-1}), z(t_{k-1}))h^q - \sum_{j=1}^k C_j^q z(t_{k-j}), \\ w(t_k) &= f_4(u(t_{k-1}))h^q - \sum_{j=1}^k C_j^q w(t_{k-j}). \end{aligned} \quad (9)$$

In this case, the short-memory principle has not been applied. It means that the memory length L_m tends to infinity as the iterations k increases. Figure 4 shows the simulation results with commensurate fractional-order $q = 0.98$, $b = 5$, $c = 0.001$, and initial conditions $(x(t_0), u(t_0), z(t_0), w(t_0)) = (0.1, 0.2, 0.3, 0.4)$. Since there is no equilibrium points in system (7) the resulting chaotic attractors are hidden.

V. PHYSICAL IMPLEMENTATION ON ARM-BASED BOARDS

The embedded implementation of fractional-order memristive system is developed on an ARM platform (SoC

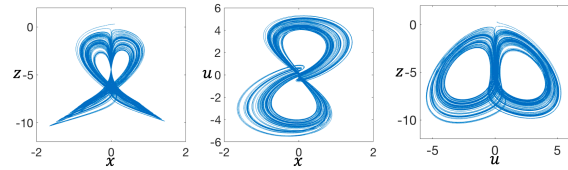


Fig. 4. Hidden chaotic attractors of the fractional-order memristive system (7) obtained with a memory length $L_m = k$ and $h = 0.01$.

BCM2387) as shown in Fig. 5. The C programming language is chosen for the set-up due to its highly portable and widely used in many operating systems.

By considering the proposed algorithm in Fig. 3 for efficient computation of the short-memory principle, the steps for implementing fractional-order systems are as follows. First, the input data are read. This includes the initialization of parameters values as well as the simulation time t_k which is computed indefinitely for real engineering applications. Then, a memory window M for the Grünwald-Letnikov method must be defined prior the iterative process. In this manner, M defines the dimension of the vector $\mathbf{Aux}$. Previous solutions are deleted since the vector $\mathbf{Aux}$ has always a fixed dimension. Next, the solution of system (7) are continuously computed with a proper memory window M . Finally, the output variables (x, u, z) are observed in an oscilloscope by using a general purpose I/O port of the ARM board, and external 8-bits DACs. Due to the output variables are digital, a post-processing technique to round and offset the output values is needed. This post-processing does not affect the computed solution since there is no feedback from analog output to digital input. Figure 6 shows the experimental results of the fractional-order memristive system (7) by using different sizes for the memory window M in the proposed approach given in section III. As can be seen from Fig. 6, the smaller the size of M , the smaller the dimension of auxiliary vector $\mathbf{Aux}$ in Fig. 3. It means that the use of computational memory of the embedded hardware (e.g. a RAM memory) can be optimized as shown in Fig. 7. For that case, we compare the required hardware resources for memory windows $M = 100$ and $M = 100,000$. In the column "VIRT" of Fig. 7, we can observe the total amount of computational memory consumed in kB. As expected, the computational memory consumption

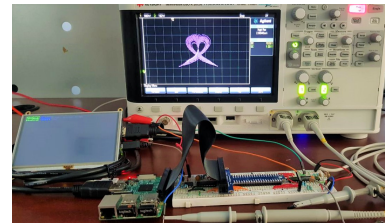


Fig. 5. Proposed experimental setup for ARM SoC digital implementation of Fractional-order Memristive Hyperchaotic System.

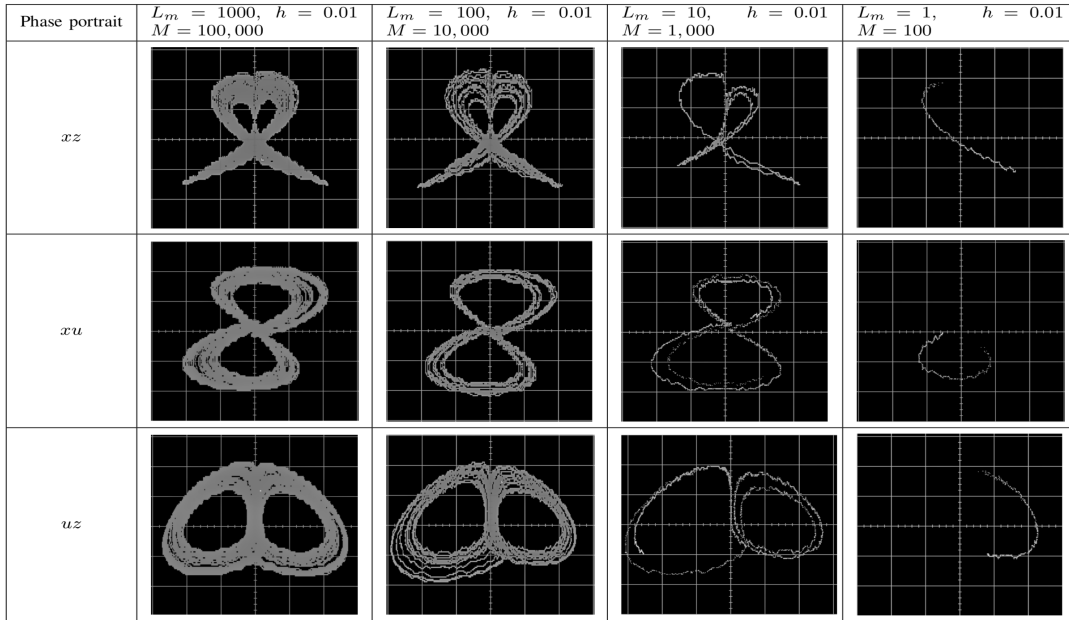


Fig. 6. Experimental results obtained from the ARM-based implementation using the proposed approach of section III. By reducing the window size M lesser information from hyperchaotic attractor is visualized on the oscilloscope.

of the embedded implementation in ARM board is reduced for short-size memory windows M , and therefore requires less hardware resources.

PID USER	PR	NI	VIRT	PID USER	PR	NI	VIRT
3915 pi	20	0	2972	3959 pi	20	0	6752
(a)				(b)			

Fig. 7. Computational memory consumptions of proposed algorithm on the ARM platform for the memristive system (7): a) Memory window $M = 100$, b) Memory window $M = 100,000$.

VI. CONCLUSION

The physical implementation of a 4D- fractional-order memristive system with hidden attractors based on ARM SoC platform has been presented. By introducing an efficient computation of the short-memory principle in the Grünwald-Letnikov method, the size of the solution vector is optimized, and therefore the required hardware resources for embedded realizations were reduced as showed herein. The experimental results were consistent with theoretical formulations. This work could open new areas for developing ARM platforms-based applications using fractional-order chaotic systems. Although the algorithm presented here is designed for sequential data processing, it could be adjusted to perform parallel computing by using FPGA platforms.

ACKNOWLEDGMENT

D. Clemente-López acknowledges CONACYT for M.Sc. scholarship no. 629054. J.M. Munoz-Pacheco acknowledges

CONACYT for the projects No. 258880 (Proyecto Apoyado por el Fondo Sectorial de Investigacion para la Educacion); VIEP-BUAP 2019; Plan de Trabajo CA (BUAP-CA-276).

REFERENCES

- [1] J.M. Muñoz-Pacheco, E. Zambrano-Serrano, C. Volos, Sajd Jafari, J. Kengne and K. Rajagopal, "A new fractionanl order chaotic system with different families of hidden and self-excited attractors," *Entropy*, vol 20(8), pp .564, 2018.
- [2] O. I. Tacha, J. M. Muñoz-Pacheco, E. Zambrano-Serrano, L. N. Stoubolus and V. T. Pham, "Determining the chotic behavior in a fractional-order finance system with negative parameters," *Nonlinear Dynamics*, vol 94(2), pp. 1303-1317, 2018.
- [3] K. Rajagopal, A. Karthikeyan, P. Duraisamy and R. Werdegiordis, "Bifurcation and Chaos in Integer and Fractional Order Two-Degree-of-Freedom Shape Memory Alloy Oscillators," *Complexity*, vol. 2018, 2018.
- [4] Q. Mo, Y. Simin, W. Yuqiong, L. Jinhu, H. Jiabin and L. Zhuosheng, "Design and FPGA Implementation of a Universal Chaotic Signal Generator Based on the Verilog HDL Fixed-Point Algorithm and State Machine Control," *International Journal of Bifurcation and Chaos*, vol 27(3), 2017.
- [5] M. F. Tolba, A. M. AbdelAty, N. S. Soliman, L. A. Said, A. H. Madian, A. T. Azar and A. G. Radwan, "FPGA implementation of two fractional order chaotic systems," *International Journal of Electronics and Communications*, vol 78, pp. 162-172, 2017.
- [6] E. Zambrano-Serrano, J. M. Muñoz-Pacheco and E. Campos-Cantón, "Chaos generation in fractional-order switched systems and its digital implementation, *International Journal of Electronics and communications*," vol 79, pp. 43-52, 2017.
- [7] Y. Xu, and Z. He, "The short memory principle for solving Abel differential equation of fractional order," *Computers and Mathematics with Applications*, vol 62, pp. 4796-4805, 2011.
- [8] C. Volos, V. T. Pham, E. Zambrano-Serrano, J.M. Muñoz-Pacheco, S. Vaidyanathan and E. Tlelo-Cuautle, "Analysis of a 4-D Hyperchaotic Fractional-Order Memristive System with Hidden Attractors," *Advances in Memristors, Memristive Devices and Systems*, pp. 207-235, 2017.

ARISTOTLE UNIVERSITY OF THESSALONIKI
Physics Department – Section of Applied & Environmental Physics
Laboratory of Nonlinear Systems, Circuits & Complexity
54124 THESSALONIKI, GREECE



Christos K. Volos
Assistant Professor
Tel. +30 2310 998284, FAX: +30 2310 998284,
e-mail: volos@physics.auth.gr

Thessaloniki, April 5, 2019

Re: Letter for Mr. Daniel Clemente-Lopez

National Council of Science and Technology (CONACYT)
Av. Insurgentes Sur 1582, Col. Crédito Constructor
Del. Benito Juárez, México, D.F.
C.P.: 03940, MEXICO.

To who it may concerns

Dear Sir,

I want to inform you that Daniel Clemente-Lopez, ID: 217470648, alumnus of the Facultad de Ciencias de la Electrónica at Autonomous University of Puebla (BUAP), Mexico, has been accepted to perform an academic internship under my co-supervision from May 1th, 2019 to June 30, 2019.

During this period, Daniel Clemente-Lopez will be officially enrolled in the Laboratory of Nonlinear Systems - Circuits & Complexity (LaNSCom), Department of Physics, Aristotle University of Thessaloniki, Greece.

Please, do not hesitate contact me for further information.

Sincerely yours,

A handwritten signature in blue ink, appearing to read 'Christos Volos', written over a horizontal line.

Dr Christos Volos
Assistant Professor

Reporte de Actividades de la estancia académica

H. Puebla de Zaragoza, 5 de
julio de 2019

El que suscribe Ing. Daniel Clemente López, por medio del presente hago informe de las actividades realizadas en la estancia académica, realizada en el periodo del 1° de mayo al 30 de junio de 2019.

Nombre y CVU del becario: Daniel Clemente López, 859036.

No. de becario: 629054.

Institución de origen: Benemérita universidad Autónoma de Puebla.

Nombre del programa de posgrado: Maestría en Ciencias de la Electrónica Opción en Automatización.

Institución receptora: Universidad Aristóteles de Salónica, Salónica, Grecia

Fecha de inicio y término de la beca de movilidad otorgada: 1° de mayo al 30 de junio de 2019.

Título del proyecto de investigación: Implementación de sistemas caóticos de orden fraccional en plataformas FPGA.

Objetivos y actividades propuestos para la estancia:

El objetivo de este proyecto es diseñar estrategias para la realización digital de sistemas caóticos de orden fraccional, de esta manera empleamos Grünwald-Letnikov (GL) y el método de descomposición de Adomian (ADM) para resolver las ecuaciones diferenciales de orden fraccional. El objetivo principal consiste en diseñar esquemas eficientes para implementar ambos métodos (GL y ADM) en plataformas FPGA.

De forma específica, los sistemas caóticos de orden fraccional a estudiar son los siguientes:

- a) Sistema de Chen.
- b) Sistema de multiples enrollamientos de orientación 1-D y 2-D.

El cronograma de actividades de presenta en la siguiente tabla:

Actividades	Abril	Junio
1. Caracterización de los sistemas de orden fraccional mediante simulaciones numéricas elaboradas en MATLAB.	X	
2. Diseñar esquemas para la implementación de los métodos de ADM y GL en FPGA.	X	
3. Analizar la compensación entre precisión y costo de hardware.	X	

4. Validar los resultados experimentales mediante la obtención de mapas de Poincaré y exponentes máximos de Lyapunov.		X
5. Estudiar varios sistemas caóticos de orden fraccional de acuerdo a los esquemas diseñados en este trabajo, mediante simulaciones numéricas en MATLAB.		X
6. Preparar un borrador de un artículo de revista.		X

Avances y descripción de los objetivos comprometidos.

1. Los resultados obtenidos mediante caracterización de sistemas caóticos mediante simulaciones numéricas en MATLAB ha permitido observar el régimen caótico de los sistemas designados para este trabajo. Este primer paso ha servido como punto de validación para los resultados experimentales.

2. La implementación de sistemas caóticos de orden fraccional se ha desarrollado en una tarjeta de desarrollo DE0-Nano, esta tarjeta cuenta con un FPGA Cyclone IV EP4CE22F17C6N, para analizar la señal caótica generada por el FPGA se ha utilizado un convertidor digital-analógico de 16 bits (AD569), finalmente el retrato de fase es capturado de forma aleatoria mediante un osciloscopio. En la fig. 1 se presenta la configuración experimental que se ha utilizado para la implementación de sistemas caóticos.

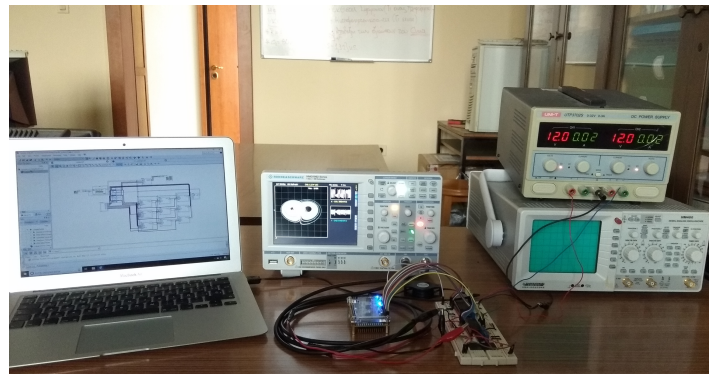


Fig. 1 Configuración experimental para la implementación de sistemas caóticos.

En la fig. 2 se ilustran los pasos que se han utilizado en el diseño digital en FPGA para la implementación de sistemas caóticos de orden fraccional. Primero se introducen los datos del sistema, esto incluye el tiempo discreto t_k durante el cual se ejecutara el cálculo computacional, el cual es indefinido para aplicaciones de ingeniería. Lo siguiente es realizar el computo iterativo para obtener la solución del sistema (en esta sección se embebe el método numérico a utilizar ADM o GL), lo siguiente consiste en la ejecución de dos procesos de forma simultánea, por una parte, la solución obtenida requiere un procesamiento para redondear y compensar los valores de salida digitales antes de su conversión D/A, por otra

parte, la solución obtenida sirve como retroalimentación para el siguiente proceso de computo.

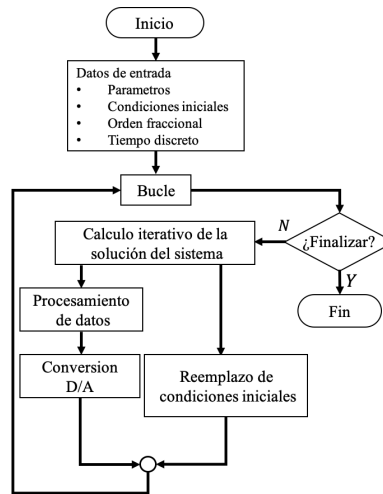


Fig. 2 Diagrama de flujo para la implementación de sistemas caóticos de orden fraccional en FPGAs.

3. Durante este trabajo se han observado los siguientes retos para la implementación de sistemas caóticos de orden fraccional

Grünwald-Letnikov

- Requiere de un alto número de localidades de memoria para obtener la solución del sistema.
- Este método paga un precio en precisión al reducir el número de localidades.
- La implementación de este método requiere de la reestructuración del algoritmo mediante el uso de contadores adicionales, estos evitan el desborde de memoria de hardware.

Método de descomposición de Adomian

- La complejidad de la implementación digital depende del tipo de no linealidad del sistema.
- La convergencia de este método incrementa en complejidad para el caso de sistemas altamente sensibles a perturbaciones, e.g. sistemas que presenten multiestabilidad.

El diseño digital en FPGA basado en el método de Grünwald-Letnikov permite una generalización en la implementación para una familia de sistemas caóticos de orden fraccional, no obstante, al requerir el almacenamiento de estados previos del sistema, la demanda computacional puede incrementar o disminuir dependiendo del sistema o familia que se requiera implementar.

ADM aproxima la solución de un sistema caótico de orden fraccional desde el punto de vista de memoria completa, requiriendo una localidad de memoria para almacenar la solución del

sistema, pese a que la implementación de este método es de caso particular para cada sistema caótico. La demanda computacional se mantiene fija independientemente del orden fraccional que se requiera implementar.

4. En la validación de los resultados experimentales, se ha observado que los resultados por la implementación de sistemas caóticos de orden fraccional muestran consistencia con los resultados obtenidos por simulaciones numéricas. Los mapas de Poncaire y exponentes de Lyapunov obtenidos corroboran el régimen caótico de la realización experimental.

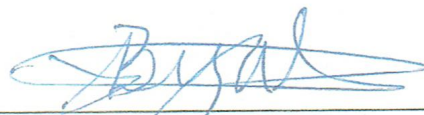
5. Mediante el uso de simulaciones numéricas se determinó que los esquemas desarrollados en el presente trabajo pueden ser aplicados a varios sistemas caóticos de orden fraccional.

6. Se está desarrollando un artículo en base a los logros obtenidos durante la estancia académica.



Dr. Jesús Manuel Muñoz Pacheco
Tutor

Facultad de Ciencias de la Electrónica
Benemérita universidad Autónoma de Puebla
(BUAP), Av. San Claudio y 18 sur. C.P. 72570.
Puebla, México
Teléfono: (+52) 2222 295500
Correo: jesusm.pacheco@correo.buap.mx



Dr. Christos Volos
Co tutor

Laboratory of Nonlinear Systems –
Circuits & Complexity (LaNSCom)
Physics Department
Aistotle University of Thessaloniki
GR, 54124, Thessaloniki, Greece.
Teléfono: (+30) 2310 998284
Correo: volos@physics.auth.gr



Ing. Daniel Clemente López
Becario

Facultad de Ciencias de la Electrónica
Benemérita universidad Autónoma de Puebla
(BUAP), Av. San Claudio y 18 sur. C.P. 72570.
Puebla, México
Teléfono: (+52) 9382 551504
Correo: ironac77@gmail.com

Bibliografía

- [1] Edward Lorenz. *The Essence of Chaos*. University of Washington Press, 1995.
- [2] José MV Grzybowski, Marcio Eisencraft, and Elbert EN Macau. Chaos-based communication systems: Current trends and challenges. In *Applications of Chaos and Nonlinear Dynamics in Engineering-Vol. 1*, pages 203–230. Springer, 2011.
- [3] Nathan Holt. Chaotic cryptography: Applications of chaos theory to cryptography. 2017.
- [4] Bingjie Wang, Hang Xu, Peng Yang, Li Liu, and Jingxia Li. Target detection and ranging through lossy media using chaotic radar. *Entropy*, 17(4):2082–2093, 2015.
- [5] Xue Zhou, Su Yang Li, and You Jie Ma. Chaos control and its application in power systems. *Applied Mechanics and Materials*, 316-317:60–63, 04 2013.
- [6] Xizhe Zang, Sajid Iqbal, Yanhe Zhu, Xinyu Liu, and Jie Zhao. Applications of chaotic dynamics in robotics. *International Journal of Advanced Robotic Systems*, 13, 03 2016.
- [7] Lei Zhang. Hénon map chaotic system analysis and vhdl-based fixed-point fpga implementation for brain stimulation. pages 1–4, 04 2017.
- [8] Shu Dai and David G Schaeffer. Chaos for cardiac arrhythmias through a one-dimensional modulation equation for alternans. *Chaos (Woodbury, N.Y.)*, 20:023131, 06 2010.
- [9] J. M. Munoz-Pacheco and E. Tlelo-Cuautle. Synthesis of n-scroll attractors using saturated functions from high-level simulation. *Journal of Physics: Conference Series*, 96:012050, 2008.

- [10] A.-A. Khennaoui, A. Ouannas, S. Bendoukha, G. Grassi, R. Pierre-Lozi, and V.-T. Pham. On fractional-order discrete-time systems: Chaos, stabilization and synchronization. *Chaos, Solitons and Fractals*, 119:150–162, 2019.
- [11] M Sasso, G Palmieri, and D Amodio. Application of fractional derivative models in linear viscoelastic problems. *Mechanics of Time-Dependent Materials*, 15(4):367–387, 2011.
- [12] Mustafa Dursun and Elif Kaşifoğlu. Design and implementation of the fpga-based chaotic van der pol oscillator. *Advanced Researches and Engineering Journal*, 02:309–314, 2018.
- [13] Viet-Thanh Pham, Christos Volos, Sajad Jafari, Zhouchao Wei, and Xiong Wang. Constructing a novel no-equilibrium chaotic system. *International Journal of Bifurcation and Chaos*, 24, 01 2014.
- [14] Chunyan Han and Simin Yu. Generating chaos via nonlinear system switching anti-control and circuit implementation. *Optik - International Journal for Light and Electron Optics*, 125:3491–3498, 07 2014.
- [15] José Tenreiro Machado, Virginia Kiryakova, and Francesco Mainardi. Recent history of fractional calculus. *Communications in Nonlinear Science and Numerical Simulation - COMMUN NONLINEAR SCI NUMER SI*, 16:1140–1153, 03 2011.
- [16] Dorčák úbomír, Jan Terpak, Ivo Petráš, and Dorčáková Františka. Electronic realization of the fractional-order systems. *Acta Montanistica Slovaca*, 12, 10 2007.
- [17] Donato Cafagna and Giuseppe Grassi. Chaos in a new fractional-order system without equilibrium points. *Communications in Nonlinear Science and Numerical Simulation*, 19:2919–2927, 2014.
- [18] Mo Qiu, Simin Yu, Yuqiong Wen, Jinhu Lü, Jianbin He, and Zhuosheng Lin. Design and fpga implementation of a universal chaotic signal generator based on the verilog hdl fixed-point algorithm and state machine control. *International Journal of Bifurcation and Chaos*, 27, 03 2017.

- [19] E Zambrano-Serrano, JM Munoz-Pacheco, and E Campos-Cantón. Chaos generation in fractional-order switched systems and its digital implementation. *AEU-International Journal of Electronics and Communications*, 79:43–52, 2017.
- [20] Ummu Mohd Roslan, Zabidin Salleh, and Adem Kilicman. Solving zhou chaotic system using fourth-order runge-kutta method. *World Applied Sciences Journal*, 21:939–944, 01 2013.
- [21] Shichang Ma, Yufeng Xu, and Wei Yue. Numerical solutions of a variable-order fractional financial system. *Journal of Applied Mathematics*, 2012:14, 2012.
- [22] Mohsen Zayernouri and Anastasios Matzavinos. Fractional adams-bashforth/moulton methods: An application to the fractional keller-segel chemotaxis system. *Journal of Computational Physics*, 317:1–14, 2016.
- [23] Mohammed F Tolba, Amr M AbdelAty, Nancy S Soliman, Lobna A Said, Ahmed H Madian, Ahmad Taher Azar, and Ahmed G Radwan. Fpga implementation of two fractional order chaotic systems. *AEU-International Journal of Electronics and Communications*, 78:162–172, 2017.
- [24] Ana Dalia Pano-Azucena, Esteban Tlelo-Cuautle, Jesus Manuel Muñoz-Pacheco, and Luis Gerardo de la Fraga. Fpga-based implementation of different families of fractional-order chaotic oscillators applying grünwald–letnikov method. *Communications in Nonlinear Science and Numerical Simulation*, 72:516–527, 2019.
- [25] Zhouchao Wei, Karthikeyan Rajagopal, Wei Zhang, Sifeu Takougang Kingni, and Akif Akgül. Synchronisation, electronic circuit implementation, and fractional-order analysis of 5d ordinary differential equations with hidden hyperchaotic attractors. *Pramana*, 90(4):50, 2018.
- [26] Shaobo He, Kehui Sun, and Huihai Wang. Dynamics of the fractional-order lorenz system based on adomian decomposition method and its dsp implementation. *IEEE/CAA Journal of Automatica Sinica*, 2016.
- [27] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC Press, 2018.
- [28] Vadim S Anishchenko, Vladimir Astakhov, Alexander Neiman, Tatjana Vadivsova, and Lutz Schimansky-Geier. *Nonlinear dynamics of chaotic and stochastic*

- systems: tutorial and modern developments*. Springer Science & Business Media, 2007.
- [29] Domine Leenaerts and Wim MG Van Bokhoven. *Piecewise linear modeling and analysis*. Springer Science & Business Media, 2013.
- [30] Joos Vandewalle and L Vandenberghe. *Piecewise-linear circuits and piecewise-linear analysis*. 2009.
- [31] Leon O Chua, Charles A Desoer, and Ernest S Kuh. *Linear and nonlinear circuits*. 1987.
- [32] Sergio Domínguez, Pascual Campoy, and José María Sebastián. *Control en el espacio de estado*. Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros, 2000.
- [33] E. Ott. *Chaos in Dynamical systems*. Cambridge University Press, 1994.
- [34] G Birta Louis and Arbez Gilbert. *Modelling and simulation: Exploring dynamic system behaviour*. *Ottawa: School of information technology and engineering*, 2007.
- [35] Hongguang Sun, Yong Zhang, Dumitru Baleanu, Wen Chen, and YangQuan Chen. A new collection of real world applications of fractional calculus in science and engineering. *Communications in Nonlinear Science and Numerical Simulation*, 64, 04 2018.
- [36] Ivo Petráš. *Fractional-order nonlinear systems: modeling, analysis and simulation*. Springer Science & Business Media, 2011.
- [37] Hongguang Sun, Yong Zhang, Dumitru Baleanu, Wen Chen, and YangQuan Chen. A new collection of real world applications of fractional calculus in science and engineering. *Communications in Nonlinear Science and Numerical Simulation*, 64, 04 2018.
- [38] Carlos Fernández Pérez, Francisco José Vázquez Hernández, and José Manuel Vegas Montaner. *Ecuaciones diferenciales y en diferencias: sistemas dinámicos*. Editorial Paraninfo, 2003.

- [39] Varsha Daftardar-Gejji and Sachin Bhalekar. Chaos in fractional ordered liu system. *Computers & mathematics with applications*, 59(3):1117–1127, 2010.
- [40] Peter A Cook. *Nonlinear dynamical systems*. Prentice Hall International (UK) Ltd., 1994.
- [41] Luis Lara, César Stoico, Rodrigo Machado, and Mario Castagnino. Estimación de los exponentes de lyapunov. *Revista Argentina: Mecánica Computacional*, 12, 2003.
- [42] Michael T Rosenstein, James J Collins, and Carlo J De Luca. A practical method for calculating largest lyapunov exponents from small data sets. *Physica D: Nonlinear Phenomena*, 65(1-2):117–134, 1993.
- [43] ÖZER Ahmet and AKIN Erhan. Tools for detecting chaos. *Sakarya University Journal of Science*, 9(1):60–66, 2005.
- [44] Angelo Vulpiani. *Chaos: from simple models to complex systems*, volume 17. World Scientific, 2010.
- [45] Fernando Reyes Cortes. *Matlab aplicado a robótica y mecatrónica*. Number 681.51 670.4272. 2012.
- [46] Carlos Mata Rodríguez. Análisis comparativo de los métodos de euler y runge-kutta en la solución numérica de ecuaciones diferenciales de primer orden mediante programación en mathcad. *Revista Ingeniería, Matemáticas y Ciencias de la Información*, 3(5), 2016.
- [47] Marius-F Danca, Wallace KS Tang, Qingyun Wang, and Guanrong Chen. Suppressing chaos in fractional-order systems by periodic perturbations on system variables. *The European Physical Journal B*, 86(3):79, 2013.
- [48] Yufeng Xu and Zhimin He. The short memory principle for solving abel differential equation of fractional order. *Computers & Mathematics with Applications*, 62(12):4796–4805, 2011.
- [49] Hossein Jafari and Varsha Daftardar-Gejji. Solving a system of nonlinear fractional differential equations using adomian decomposition. *Journal of Computational and Applied Mathematics*, 196(2):644–651, 2006.

- [50] Donato Cafagna and Giuseppe Grassi. Fractional-order chua's circuit: time-domain analysis, bifurcation, chaotic behavior and test for chaos. *International Journal of Bifurcation and Chaos*, 18(03):615–639, 2008.
- [51] Andrew Sloss, Dominic Symes, and Chris Wright. *ARM system developer's guide: designing and optimizing system software*. Elsevier, 2004.
- [52] Santanu Kundu and Santanu Chattopadhyay. *Network-on-chip: the next generation of system-on-chip integration*. CRC press, 2018.
- [53] R Chandana, S Jilani, and S Javeed Hussain. Smart surveillance system using thing speak and raspberry pi. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(7):214–218, 2015.
- [54] R Rani, S Lavanya, and B Poojitha. Iot based home security system using raspberry pi with email and voice alert. *International Journals of Advanced Research in Computer Science and Software Engineering ISSN*, pages 119–123.
- [55] Jinhu Lü and Guanrong Chen. A new chaotic attractor coined. *International Journal of Bifurcation and chaos*, 12(03):659–661, 2002.
- [56] Guoyuan Qi, Guanrong Chen, Shengzhi Du, Zengqiang Chen, and Zhuzhi Yuan. Analysis of a new chaotic system. *Physica A: Statistical Mechanics and its Applications*, 352(2-4):295–308, 2005.
- [57] Pei Yu, Weiguang Yao, and Guanron Chen. Analysis on topological properties of the lorenz and the chen attractors using gcm. *International Journal of Bifurcation and Chaos*, 17(08):2791–2796, 2007.
- [58] DK Guevara-Flores, JM Muñoz-Pacheco, E Zambrano-Serrano, O Félix-Beltrán, and CK Volos. Pwl function-based model of a beta cell. In *2016 5th International Conference on Modern Circuits and Systems Technologies (MOCASST)*, pages 1–4. IEEE, 2016.
- [59] Danca Marius-F, MA Aziz-Alaoui, and Small Michael. A new piecewise linear chen system of fractional-order: Numerical approximation of stable attractors. *Chinese Physics B*, 24(6):060507, 2015.

- [60] Weihua Deng and Jinhu Lü. Design of multidirectional multiscroll chaotic attractors based on fractional differential systems via switching control. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 16(4):043120, 2006.
- [61] Tao Yang and Leon O Chua. Piecewise-linear chaotic systems with a single equilibrium point. *International Journal of Bifurcation and Chaos*, 10(09):2015–2060, 2000.
- [62] Eric Campos-Cantón, Juan Gonzalo Barajas-Ramirez, G Solis-Perales, and Ricardo Femat. Multiscroll attractors by switching systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(1):013116, 2010.
- [63] Eric Campos-Cantón, Ricardo Femat, and Guanrong Chen. Attractors generated from switching unstable dissipative systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(3):033121, 2012.
- [64] Ernesto Zambrano-Serrano, Eric Campos-Cantón, and Jesús Manuel Muñoz-Pacheco. Strange attractors generated by a fractional order switching system and its topological horseshoe. *Nonlinear Dynamics*, 83(3):1629–1641, 2016.
- [65] Rodolfo de Jesús Escalante-González, Eric Campos-Cantón, and Matthew Nicol. Generation of multi-scroll attractors without equilibria via piecewise linear systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(5):053109, 2017.
- [66] Abdul-Majid Wazwaz. A new algorithm for calculating adomian polynomials for nonlinear operators. *Applied Mathematics and computation*, 111(1):33–51, 2000.
- [67] Donato Cafagna and Giuseppe Grassi. Adomian decomposition method as a tool for numerical studying multi-scroll hyperchaotic attractors. *Proc. NOLTA2005, Oct*, 2005.
- [68] Thomas Schreiber Rainer Hegger, Holger Kantz. Tisean, nonlinear time series analysis. [urlhttps://www.pks.mpg.de/tisean/](https://www.pks.mpg.de/tisean/), 2007. Accedido 03-09-2019.